1982

Deterministic aggragated model of STAR on the apple computer (DAMSTAC)

McKenzie, Charles J.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DETERMINISTIC AGGREGATED MODEL OF STAR
ON THE APPLE COMPUTER (DAMSTAC)

by

Charles J. McKenzie III

and

Thomas Joseph Pawlowski III

March 1982

Thesis Advisor: J. G. Taylor

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Deterministic Aggregated Model of STAR on the Apple Computer (DAMSTAC)	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis, March 1982 / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Charles J. McKenzie III; Thomas Joseph Pawlowski III	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	12. REPORT DATE: March 1982 / 13. NUMBER OF PAGES: 178
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Army — Apple II Computer

STAR — Lanchester-Type Aggregated Combat Model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis investigates the estimation of Lanchester attrition-rate coefficients in a deterministic aggregated combat model. Numerical values for the attrition-rate coefficients in the Lanchester-type model are taken to be the same as those for a corresponding continuous-time Markov-chain model and maximum likelihood estimators are developed for these Markov-chain coefficients. A discussion of the historical background of Lanchester's equations preceeds the theoretical

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

development of components of the attrition rate equations. Using the functional forms and procedures developed by Gordon M. Clark in his doctoral thesis, a simple computer simulation program is developed for a short duration battle consisting of homogeneous forces. Recommendations are made to modify this program to model heterogeneous forces in a similar battle. A discussion of the theoretical background supports these recommendations.

Approved for public release, distribution unlimited

Deterministic Aggregated Model of STAR on the Apple Computer (DAMSTAC)

by

Charles J. McKenzie III
Captain, United States Army
B.S., Clemson University, 1971

and

Thomas J. Pawlowski III
Captain, United States Army
B.S., United States Military Academy, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March, 1982

ABSTRACT

This thesis investigates the estimation of Lanchester attrition-rate coefficients in a deterministic aggregated combat model. Numerical values for the attrition-rate coefficients in the Lanchester-type model are taken to be the same as those for a corresponding continuous-time Markov-chain model, and maximum likelihood estimators are developed for these Markov-chain coefficients. A discussion of the historical background of Lanchester's equations preceeds the theoretical development of components of the attrition rate equations. Using the functional forms and procedures developed by Gordon M. Clark in his doctoral thesis, a simple computer simulation program is developed for a short duration battle consisting of homogeneous forces. Recommendations are made to modify this program to model heterogeneous forces in a similar battle. A discussion of the theoretical background supports these recommendations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The purpose of this thesis is to investigate methods for estimating Lanchester attrition-rate coefficients, used in a deterministic aggregated combat model, from data generated by the detailed, high resolution Monte Carlo combat simulation model called STAR (Simulation of Tactical Alternative Responses). An additional purpose is to construct a simple Lanchester-type combat model capable of using the estimated attrition rate coefficients to predict the outcome of STAR simulations.

STAR is a model which has been developed primarily by U.S. Army students attending the Naval Postgraduate School. It has recently gained acceptance in the Army Community as a viable combat model, useful for studies such as the United States Army Mortar Study to be conducted in 1982. The current version of STAR is written in SIMSCRIPT and requires about 2 megabytes of core storage. The run time for one replication of STAR involving a company size force defending against a battalion size attacking force is approximately four CPU minutes on the IBM 3033 computer.

The basic conceptual difficulties found in this thesis are summarized in Figure 1. The numerical values for the

attrition rate coefficients in the deterministic Lanchester-type model are assumed to be the same as those for a corresponding continuous-time Markov chain model. With this assumption, maximum likelihood estimators (MLE's) can be developed for the Markov chain model coefficients. Specific data needed for the MLE computations is extracted from the high resolution Monte Carlo simulation and used to compute "estimated" coefficient values. These values for the continuous time Markov chain model are then substituted into the "analogous" deterministic Lanchester-type model.

Constructing a Lanchester-type attrition model such as that proposed in this thesis is of significant value. With respect to the hierarchy of models concept [Ref. 1], it could possibly fill the requirement as a 'hook' or link between STAR and low resolution models that exist at higher levels in the pyramid of models. However, possibly the most important contribution this model can make is as a filter model for STAR to reduce costs and time to run the many variations of a scenario in STAR. To explain further, consider the upcoming mortar study. To conduct such a study many runs/replications of a battle using the STAR model will have to be made. These numerous runs are required to compare various types of mortars, various types of force mixes,

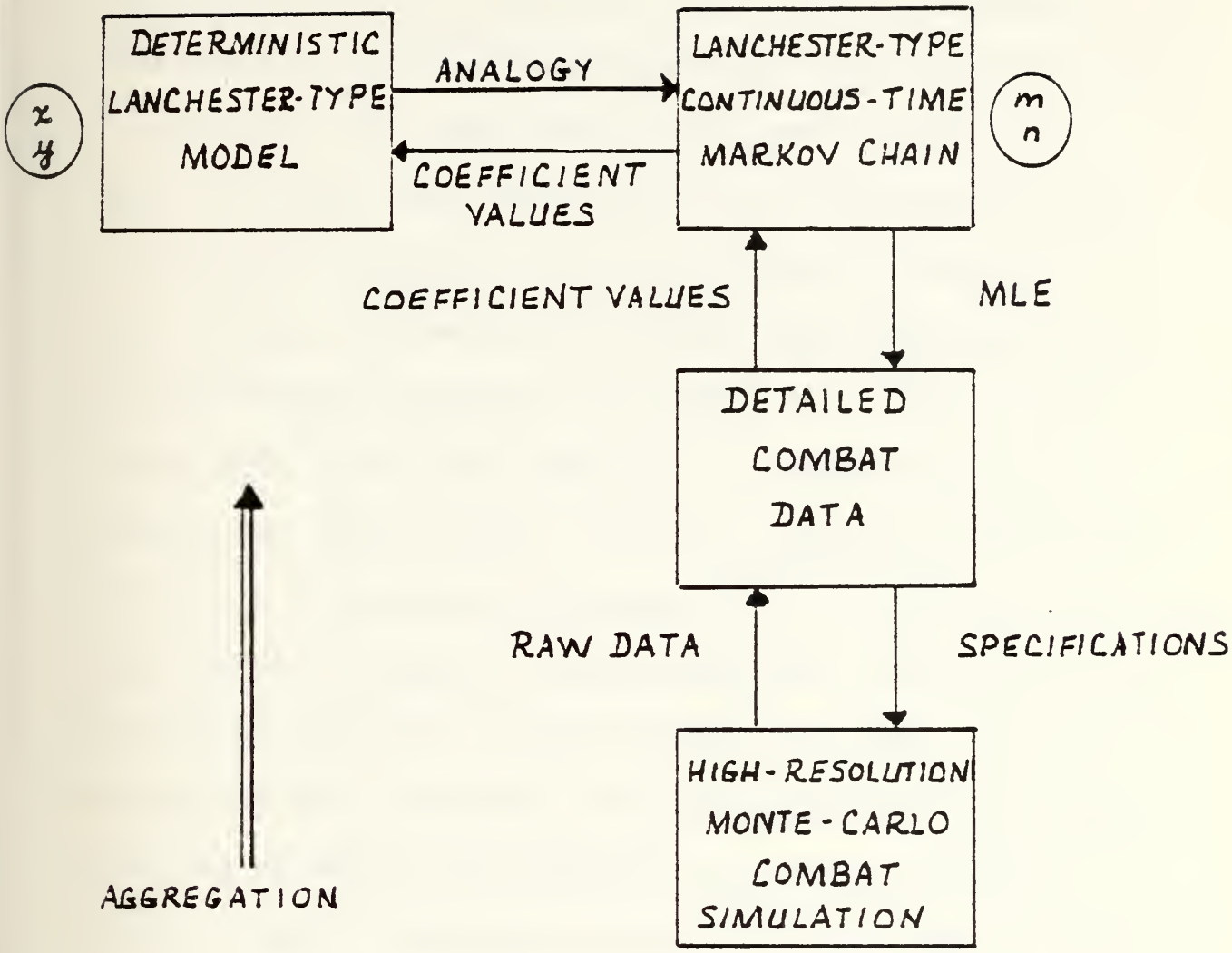


Figure 1: Relationships of Models

weapon mixes, and scenarios. Since STAR is a Monte Carlo simulation, variance-reducing replications are required.

This requires significant computer time at a significant cost. By using a simpler analytical model which produces comparable output to a STAR run, many force mixes/weapon mixes/scenarios can be eliminated from further consideration at a minimal cost. STAR can then be used to concentrate only on the more promising options for force/weapons mixes.

This thesis records the progress made in developing the small, simple, Lanchester-type combat model discussed above. It also lays the foundation for future work on this model. Chapter 2 of this thesis provides an introduction to Lanchester-type combat models. Chapter 3 outlines the development of the Deterministic Aggregated Model of STAR on the Apple computer (DAMSTAC), which represents the first step in building an analytical model comparable to STAR. Chapter 4 explores various Lanchester attrition rate functional forms which may be used in the DAMSTAC or in future generation models. Chapter 5 details some desired enhancements of the model, while Chapter 6 discusses the work required to transition from the current homogeneous force DAMSTAC model to a more realistic, albeit more complicated heterogeneous model. Chapter 7 contains the concluding remarks.

Appendix A lists the variables used in the computer program for the DAMSTAC model. Appendix B lists some notes of

interest concerning the program and model, which may not be apparent to the reader. Appendix C discusses the utility programs developed to support the model. Appendix D is a discussion of additional work done by the authors in the form of a computer model that explicitly plays the terrain effect. Appendix E consists of work done by two other students on a postprocessor which supports the computational requirements of the model and acts as an interface between STAR and the model. Finally, Appendix F contains the computer listings of all programs written for this thesis.

II. HISTORICAL BACKGROUND

Lanchester-type combat models are analytical models in which differential equations are used to express the rate of change of force levels to represent the effects of the attrition process. The concept of representing attrition effects in land warfare with such mathematical equations was originated by F. W. Lanchester in 1914 in order to quantitatively justify concentration of forces.

Lanchester hypothesized that under conditions of "modern warfare" attrition between two homogeneous forces, each using "aimed" fire, could be modelled by:

$$dx/dt = -ay(t) \quad \text{with } x(0) = x_o$$

$$dy/dt = -bx(t) \quad \text{with } y(0) = y_o$$

where $x(t)$ and $y(t)$ are the number of X and Y firers alive at time t, and a and b are positive constants that are called Lanchester attrition-rate coefficients. The constant 'a' represents the effectiveness of an x firer in killing y targets per unit of time and 'b' represents the equivalent

effectiveness of the y firers. The equations written above are usually called "Lanchester's equations for modern warfare". Clearly, these functional forms for attrition rates that Lanchester developed are directly proportional to the attrition rate coefficients and the number of firers from the opposing force. The assumptions that must be made in order to use Lanchester's aimed fire equations to model combat attrition are very restrictive. These assumptions can be stated as follows [Ref. 2]:

1. Two homogeneous forces are engaged in combat;
2. Each unit on either side is within weapon range of all units on the other side;
3. The effects of successive rounds in the target area are independent;
4. Each unit is sufficiently well aware of the location and condition of all enemy units so that it engages only live enemy units (one at a time) and kills them at a constant rate, which does not depend on the enemy force level. When an enemy target is killed, search begins for a new target, with the rate of acquiring a new enemy target being independent of the enemy force level;
5. Fire is uniformly distributed over surviving enemy units.

Lanchester also hypothesized that when both sides used "area" fire, the attrition of forces could be modelled by:

$$dx/dt = -axy$$

$$dy/dt = -bxy$$

The first three conditions under which Lanchester's equations for area fire apply are identical to those for the equations for modern warfare, but assumptions 4 and 5 are replaced by [Ref. 3]:

4. Each firing unit is aware only of the general area in which enemy forces are located and fires into this area without feedback about the consequences of its fire;
5. Fire from surviving units is uniformly distributed over the area in which enemy forces are located;
6. Each unit presents the same vulnerable area to enemy fire. This vulnerable area is much larger than the effective area of a single round of enemy fire. Additionally, the number of hits required for a kill obeys a geometric probability law.

The state equation that relates the force levels and does not explicitly contain time for the aimed-fire assumptions is known as Lanchester's Square Law,

$$b(x_o^2 - x^2) = a(y_o^2 - y^2).$$

The state equation for the area fire model yields Lanchester's Linear Law,

$$b(x_o - x) = a(y_o - y).$$

From these state equations many important results can be deduced, for example, battle outcome prediction conditions. Further analysis of Lanchester's classical equations and their implications will not be done here. Suffice it to say that Lanchester's original work forms the basis for the analytical combat model, with his simple model serving as a point of departure for the more complicated and elaborate analytical models discussed later in this thesis. Readers of this thesis and students who intend to continue with work initiated here should have a sound understanding of Lanchester's basic models before continuing further.

Lanchester's equations for modern warfare and his equations for area fire have, over the years, been thoroughly dissected and analyzed by students of differential equations and students of analytical combat models. One such student was Gordon Clark, a doctoral student at Ohio State University in the late 1960's. He was well aware of the power that simulation models such as Carmonette, a high resolution combat model, had for describing the complex process of combat. However, the major drawback to simulation, as he

perceived it, was that the method was slow and expensive. Clark proposed that the use of an additional model to interpret simulation results would greatly benefit Army studies which required numerous runs with the simulation model. He went on to develop this analytical model which he called the Combat Analysis Model (COMAN). This much simpler and faster running model would greatly reduce computer execution time and costs, and could serve as a filter model for Carmonette. A similar approach to Clark's methodology in developing COMAN will be used here to develop the analytical model for STAR. The next chapter details the first steps taken to build the analytical combat model for STAR.

III. DAMSTAC MODEL

The aggregated force-on-force analytical model is an integral part of combat modelling and serves several significant purposes. With the high cost of computer time and the limited computer core storage available to most military users, the analytical combat model offers a viable alternative to detailed high resolution simulations.

Analytical combat models can be used in a stand alone configuration or in conjunction with a high resolution simulation. In the stand alone mode, the models are usually "transparent", or, in other words, easy to understand. They are also small, quick but most importantly, they increase the user's understanding of the process of combat. In conjunction with high resolution combat simulations, the analytical model offers, as Gordon Clark states in his doctoral thesis, the opportunity to "extrapolate the results of the combat simulation to predict the outcome from force mixes that were not explicitly simulated." In this way it can act as a filter to eliminate certain force mixes from further investigation. Also, Clark identifies another use for it as "an integral component of a large unit model to predict the outcome from individual unit actions." [Ref. 4]

A major objective of this thesis was to build a simple aggregated combat model on the Apple II microcomputer. The input parameters for the model were estimated using the output data from a STAR run. This concept of estimating the attrition rate coefficients involved additional work in developing a data postprocessor compatible with STAR, and significant effort in the statistical analysis of the information provided by the postprocessor.

There are several reasons why the Apple II computer was chosen for the development of the model rather than a main frame computer. First, the need exists for a 'portable' model that is easily accessible to the Army analyst. A desk top type system that can run a model quickly and with minimum setup time will be a valuable asset to the analyst. With the Apple II computer gaining wider acceptance in the Army for various uses, the accessibility of a model on such a system is greatly increased.

Second, the nature of the model is such that it lends itself to a small system format. The aggregated form of the model reduces the required core storage from that required by a high resolution model. The concept of DAMSTAC is to serve as a filter model to weed out unacceptable force mixes or scenarios before turning to a high resolution model. The

idea here is to reduce the time and cost of running numerous iterations of a large, high resolution model on a main frame computer.

Third, the cost of a computer system to support the model is certainly economical. At a cost of under $3000, the Apple II computer system with a model such as DAMSTAC provides a powerful analytical tool for a minimal cost, certainly in relation to the cost of a main frame computer.

Fourth, the Apple II has the capability to provide additional analysis functions on the output of the model. With such programs as Apple Plot and available commercial statistical and numerical analysis packages, most of the needed analytical work can be accomplished on the Apple II computer. This can result in additional savings in time by eliminating the requirement to transfer any data from one system to another.

The laurels cast on the Apple II should not go without some discussion of its limitations. The Apple II is limited in capacity to 64 kilobytes of Random Access Memory storage space. So, essentially the model must be able to work within the amount of core storage. There are methods of working around this problem, if it arises, but this will usually create a new problem or limitation. For example,

some of the data or a part of the program can be kept on the storage diskette and put into computer memory only when it is needed. This saves space in the computer's memory but increases the run time of the model due to the time required to access the diskette to retrieve the required information.

This raises another limitation of the Apple II--its run time. Compared to a main frame computer, the Apple II is very slow. This is, however, relatively speaking since computer time is normally measured in milliseconds and nanoseconds. For the analyst waiting for the output from DAMSTAC, the model on the Apple II would take about four to ten minutes to run. This is probably acceptable for most analysts.

A. SCENARIO

The model portrays a battle between a RED tank battalion attacking a BLUE tank company in the 10 x 10 KM Fulda Gap (Keyhole) terrain. See Figure 2 for the terrain map.

The BLUE defensive positions and the RED positions along with their attack routes (coordinating points) are provided in Table 1. The RED units attack along three preplanned attack routes toward the BLUE defensive positions. Grid coordinates for unit locations and coordinating points were computed as the centroid of the elements of the unit. In

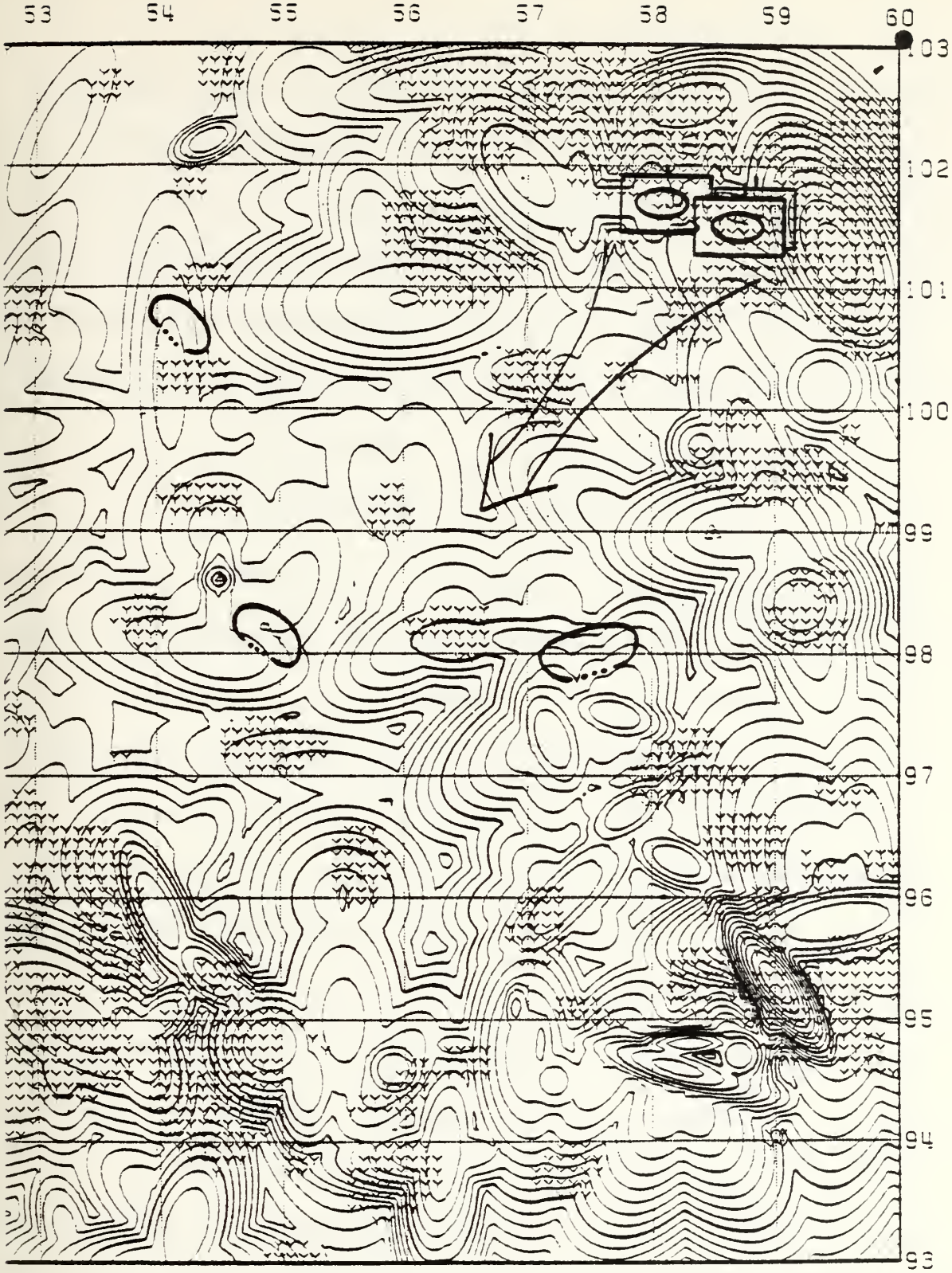


Figure 2: Terrain Map

Star's data base, each tank has its own individual location. For example, BLUE unit 1 has four different locations associated with the it, each one corresponding to a tank in the unit. The centroid of the unit was taken to be the average of these grid coordinates. Thus, the initial location for BLUE unit 1 in Table 1 is the average of these coordinates:

54150 100690	54150 100620
54190 100520	54160 100790

The rate of RED advance is a constant but specified by the user. BLUE units can move to alternate positions at a rate designated by the user. However, for this scenario, BLUE units remain stationary for the entire battle.

To obtain the raw output data that would correctly reflect the above scenario, it was necessary to "turn off" many of the submodels of STAR. For example, the Field Artillery module was shut off by scheduling the first fire mission well after the battle ended. Weapon systems other than tanks within the fighting units were rendered useless by setting their ammunition counts (basic loads) to zero. Additionally, the communications and the Air Defense modules were not played. After all of these preliminary actions were taken, the result was a battle exclusively between tanks. This effort worked well to reduce a complicated,

heterogeneous force battle to a simple homogeneous force battle. One minor problem arose, however, when a weapon system other than a tank was detected and fired on when, by the scenario, it did not even exist. The effects of this target acquisition and engagement were considered insignificant and were disregarded.

Table 1: UNIT LOCATIONS/COORDINATING POINTS

BLUE	X-COORD	Y-COORD	SIZE	REMARKS
UNIT 1	54163	100655	4	initial location
UNIT 2	54788	98120	4	initial location
UNIT 3	57448	97893	4	initial location

RED	X-COORD	Y-COORD	SIZE	REMARKS
UNIT 1	58727 52000	101491 96000	10	initial location coordinating pt
UNIT 2	58110 52000	101696 96000	10	initial location coordinating pt
UNIT 3	58727 52000	101491 96000	10	initial location coordinating pt

B. ASSUMPTIONS

The attrition rate functional form chosen for the DAMSTAC model, which will be discussed later in this thesis, requires the following assumptions.

1. The conditional kill rates, α and β , are constant and independent of time if an individual firer has at least one acquired target. If there are no acquired targets,

the kill rates are zero.

2. P and q, the probabilities of nondetection, are assumed to be constant functions of time, that they are independent of the force size and the number of acquired targets, and that the probabilities p and q are equal for all x-firer/y-target and y-firer/x-target combinations respectively.

3. Synergistic effects are neglected, therefore, the actual force attrition rate is equal to the sum of the individual kill rates for the opposing forces.

Additionally, the assumptions listed below are designed to remove the distractions of a complicated scenario and to make the initial model easier to understand:

1. All units have point locations; ie. a battalion is represented by a grid coordinate and all calculations for the battalion are made from that location.

2. Units are homogeneous. That is, they are composed of identical firing elements/weapons systems.

3. Fire is uniformly distributed over surviving enemy targets within range.

4. Sufficient supplies of ammunition for both sides exist for the duration of the battle.

5. Units move at a constant rate of speed.

C. SYNOPSIS OF THE COMPUTER MODEL

The DAMSTAC computer program is a modular type program written in the Applesoft BASIC language. The program accesses external data files as a source of input of initial data for the two forces. These data files are produced by the user with the help of the Force Maker utility program described in Appendix C. Additionally, the program has the capability to make output files that can be stored on a diskette and accessed by the user with the Results Reader utility program, also described in Appendix C. Results Reader will print the output to the screen or to a printer.

A copy of the program for the model is included as Listing A in Appendix F. A sample run of the model including user inputs is included as Figure 3. Flow charts of the main program and subroutines are included as Figure 4, Figure 6, and Figure 7.

1. Initialization of Arrays and Variables

Lines 30 thru 170 dimension the arrays and matrices and initialize hardwired data. Dimensions are set to handle up to five units in each force and up to ten coordinating points, including the initial location, for each unit. The user can modify these capacities by changing the dimensions of the appropriate variables.

```
]RUN
NAME OF FILE FOR BLUE DATA? THESIS
:
54163             100655
SIZE OF BLUE UNIT 1?4
:
54788             98120
SIZE OF BLUE UNIT 2?4
:
57448             97993
SIZE OF BLUE UNIT 3?4

NAME OF FILE FOR RED DATA? THESIS
SIZE OF RED UNIT 1?10
SIZE OF RED UNIT 2?10
SIZE OF RED UNIT 3?10

DO YOU WANT THE PRINTER ON? (Y/N) Y

TIME    BLUE          RED
0       12            30
30      12            30
60      12            30
90      12            30
120     12            30
150     12            30
180     12            30
210     12            30
240     12            30
270     12            30
300     11.3695362    29.1743717
330     10.858762     28.4987136
360     10.1687086    27.1036729
390     9.61458731    25.9203808
420     9.07784103    25.2618404
450     8.17400928    24.6831241
480     8.88690879    24.1710725
510     8.3417791     23.7152286
540     8.70492405    23.3071943
570     8.33779153    22.4180871
600     8.45823959    22.1061593
630     8.26874557    21.8237029
660     8.26874557    21.8237029
690     8.26874557    21.8237029
720     8.26874557    21.8237029
750     8.26874557    21.8237029
780     8.26874557    21.8237029
810     7.81967949    21.0730607
```

Figure 3: Sample Run of the Model

Lines 180 thru 350 read BLUE force data from a file to include the number of units, initial locations, any coordinating points, and rates of advance, and compute total force level based on user input of the unit force levels. The time increment (DT) is set to 30 seconds while the maximum time of the battle (MT) is set to 1500 seconds. These values are, of course, adjustable by changing them in line 80. For both forces, the breakpoint criteria is set at 0.3 of the force level and movement criteria is set at 0.5 of the unit strength, even though the movement criteria was not used for any runs of the model.

Lines 360 thru 520 perform the same operations for the RED forces as lines 180 thru 350 perform for the BLUE forces.

Lines 530 and 600 ask if the printer should be turned on or off and calls the printout subroutine to print the initial data.

2. Main Program (lines 610-780)

The main program performs these operations:

increments the time,

checks if the time exceeds the max time,

calls the movement and attrition subroutines,

recomputes current total force levels after attrition, and

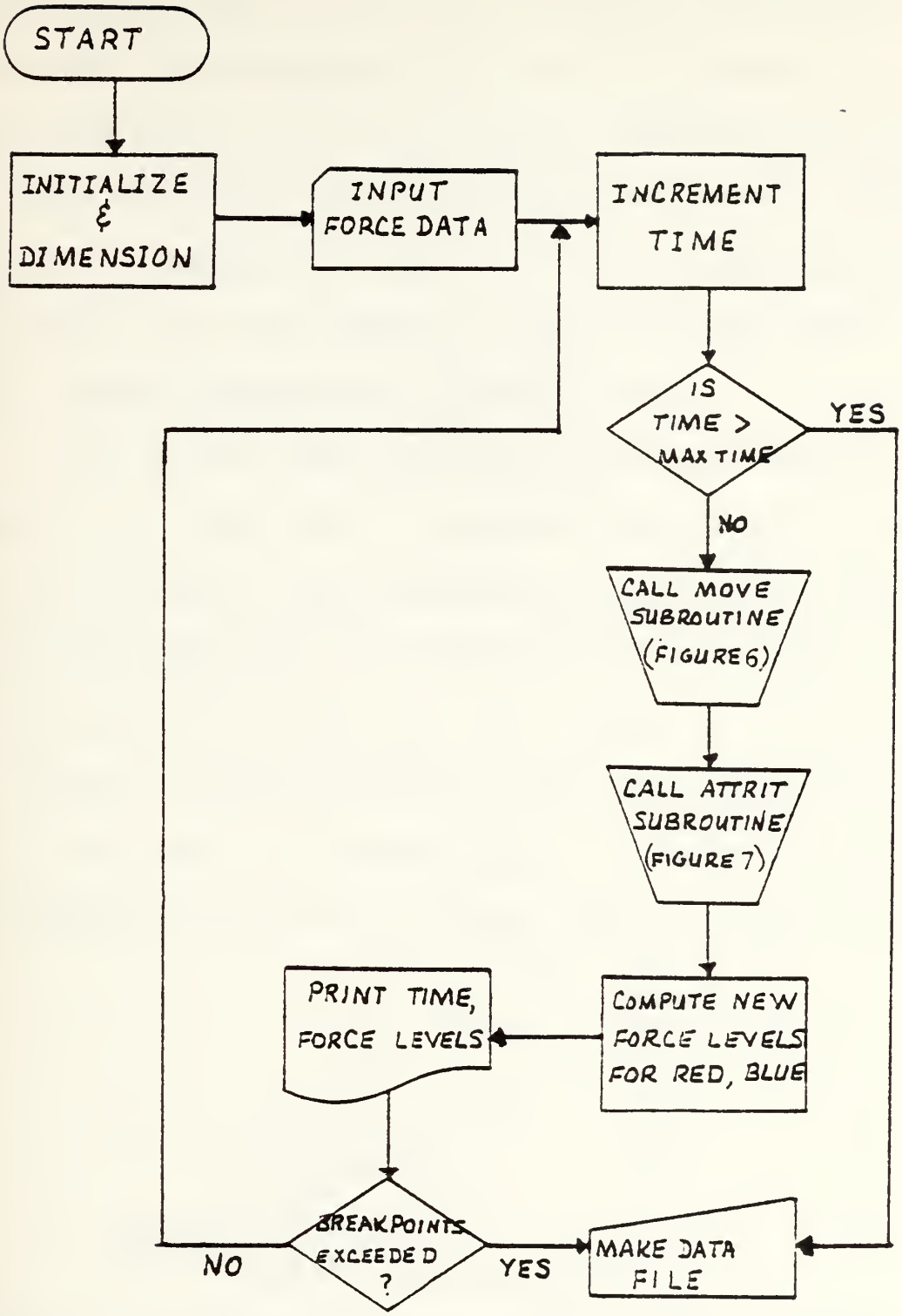


Figure 4: Flow Chart of the Main Program

checks force levels against the designated breakpoints for each force, to determine if the battle has ended.

If the breakpoints or the max time are exceeded, the battle terminates and the program prints the force levels. Additionally the program will write the force levels for each time interval onto the diskette. (lines 1470 thru 1580)

3. Movement Subroutine (lines 790-1160)

Both RED and BLUE forces are capable of moving. Movement of the BLUE force is nullified by setting its movement rate to zero in the initialization part of the program. The movement subroutine uses basic geometry to compute new coordinates for the moving unit (see Figure 5). The movement process is based on moving a unit from one coordinating point to the next in a straight line. The amount of movement depends on the rate of movement (BR(I) or RR(I)) and

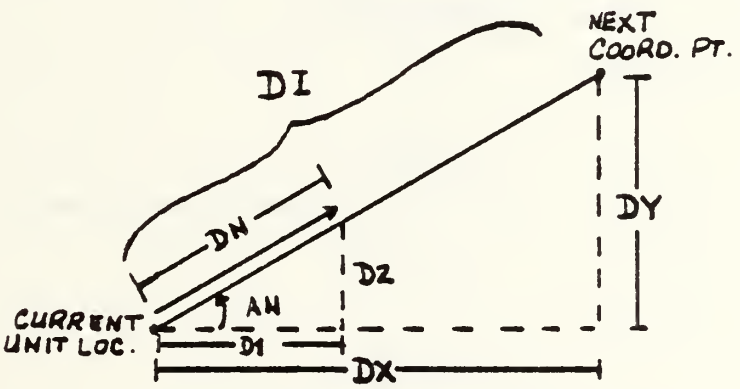


Figure 5: Geometric Representation of Unit Movement

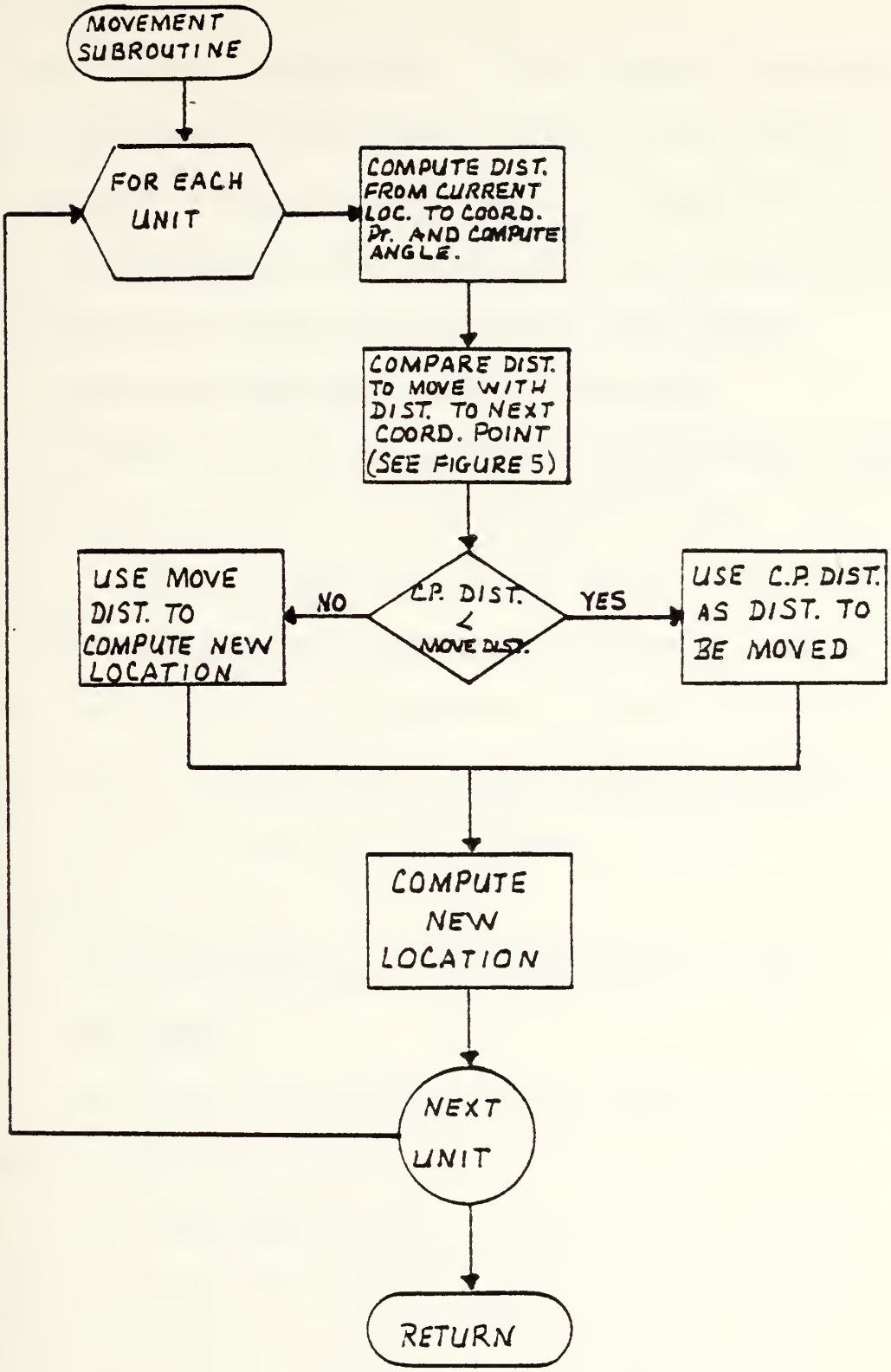


Figure 6: Flow Chart of the Movement Subroutine

the time interval length (DT). If the distance computed is farther than the next coordinating point, only movement to the coordinating point is accomplished. Lines 800 to 970 perform computations to move RED forces while Lines 980 to 1150 perform the same computations for BLUE forces.

4. Attrition Subroutine (lines 1170-1480)

A range check is made for each firer/target combination to insure the units are within firing range. If so, attrition caused by that firer on the target is computed. Next total attrition for each unit is computed based on the partial attritions of the firers as a fraction of their force levels. In other words, when a firer is firing at 3 enemy units, each target unit receives 1/3 of the firer's fire power.

5. Printout Subroutine (lines 1490-1530 & lines 1760-1940)

There are two different output formats used in the program. The first (lines 1490 to 1530) prints only a summary of the total force levels for both forces for each time interval. This is the same data that is saved to a file by the final portion of the program. The second format (lines 1760 to 1940) prints out the units of each force with their

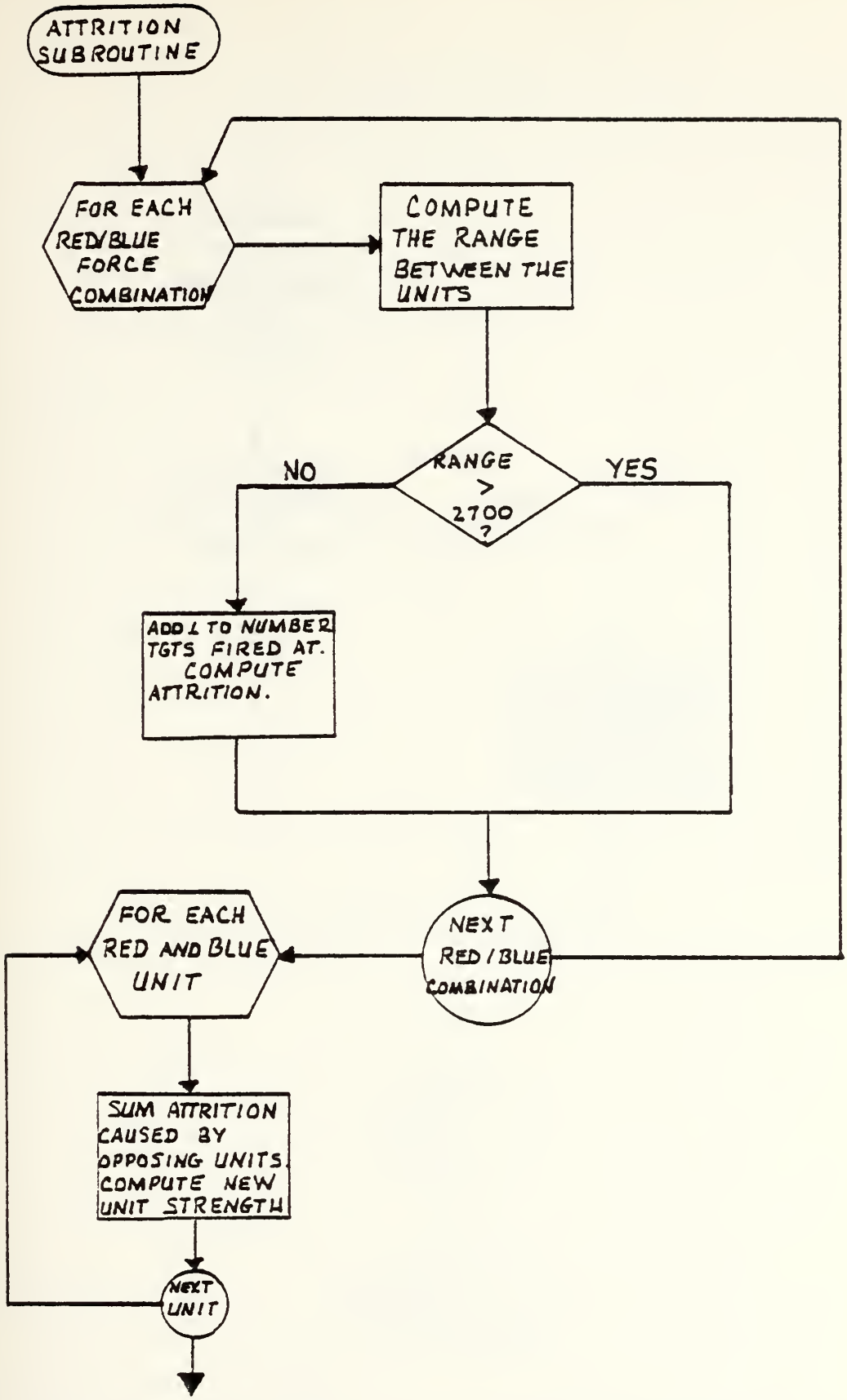
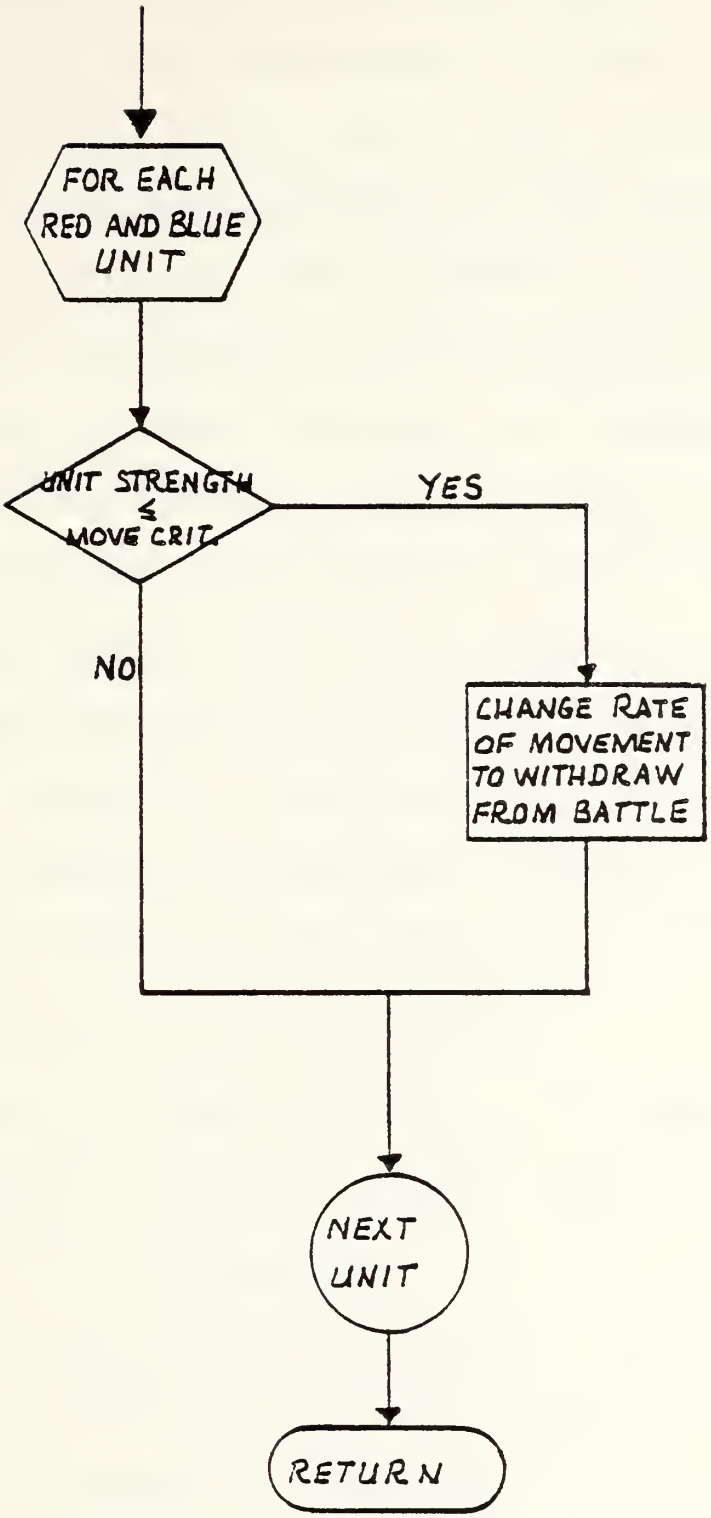


Figure 7: Flow Chart of the Attrition Subroutine


FOR EACH
RED AND BLUE
UNIT
UNIT STRENGTH
≤
MOVE CRIT.
YES
NO
CHANGE RATE
OF MOVEMENT
TO WITHDRAW
FROM BATTLE
NEXT
UNIT
RETURN

current locations and strengths. This format allows the user to see which unit, specifically, is being attritted and the grid coordinates of that unit at the time of attrition.

6. Output File Making Subroutine (lines 1540-1750)

This subroutine allows the user to make a file of the output from the current run of the program. This routine actually produces two files, one consisting of the attrition rate coefficients and probabilities of non-acquisition, and the second containing the time and force levels of both forces at that time. These files are seperated so that the second file can be used with the APPLEPLOT program to produce a graph of the two force levels as a function of time. This file has the postscript, ".RESULTS", added to its file name, while the first file with the attrition rate coefficients has the postscript ".COEFS".

D. METHODOLOGY FOR COMPUTING PARAMETER ESTIMATES

The process of building a computer simulation combat model requires the development of methodologies for modelling the various aspects of combat. Constructing a force-on-force attrition model requires methods for consolidating combat phenomena into simplified algorithms with coefficients representing one or more facets of combat. This section discusses one significant part of the

force-on-force attrition model: the methodology for estimating the attrition rate coefficients in the Lanchester type attrition rate equations.

In deriving the equations for the attrition rates (dx/dt and dy/dt), it is necessary to understand the processes that must occur to cause a firer to inflict attrition. These key processes can be summarized as:

the environmental process;

the target acquisition process;

the target engagement process.

The environmental process involves the terrain and weather factors that affect firer/target line of sight (LOS). The target acquisition process represents the visible capability of the firer to detect a target on the battlefield, while the target engagement process describes the firer's ability to hit and kill an acquired target. These three processes will all be represented by the attrition rate coefficient, A, in the attrition rate equation. This coefficient will be broken down into its three components which describe mathematically the phenomena of the three processes. The three components are:

P_V = the probability that a target is visible to the firer (environmental process)

P_a = the probability that a visible target is acquired (target acquisition process)

α = the casualty rate at which a firer attrits the acquired target (target engagement process)

α should not be confused with A, the attrition rate coefficient, of which α is a factor. The attrition rate equation may be represented in words by:

dx/dt = -[P(tgt is visible)][P(tgt is acquired)][casualty rate]

The second probability above can be broken down further to P(tgt is acquired | tgt is visible). Note that this is a conditional probability since it is assumed that the target must be visible to be acquired. By identifying the mathematical expressions for each of these components of the attrition rate coefficient, the attrition rate equation can be obtained.

1. The Environmental Process

There are generally two ways to represent line of sight in a model:

as a mathematical simulation of actual terrain;

as a stochastic process.

The mathematical simulation method normally involves a digital representation of terrain elevation over a segment of the terrain. The elevations of the segments between the

firer and the target are used to determine if line of sight exists. Virtually all high resolution models use this method, or variations of it, to determine line of sight. STAR uses a different form of mathematical simulation to represent terrain. This is called functional terrain, in which hills and forests are represented by elliptical contours of elevation. Hills are modelled by describing them in terms of bivariate normal equations with varying parameters for each hill. Forests or built-up areas are represented by an elliptical pattern on the ground again with varying parameters for each forest.

For a low resolution or highly aggregated model, the mathematical simulation method for line of sight is usually not practical or desireable in terms of core space and computation time required. Therefore, the second method, in which LOS is represented as a stochastic process, is used. The concept of this process is as follows. The effect of terrain on the process of line of sight is that a target falls into one of two categories or states: either it is visible or it is invisible.

Assuming a steady-state situation for the terrain, probabilities can be assigned to each of these two states. These probabilities represent the percentage of time that

the target will be in that state (visible or invisible) for that terrain over a reasonable amount of time. Graphically, the states of the target can be represented by the diagram in Figure 8.

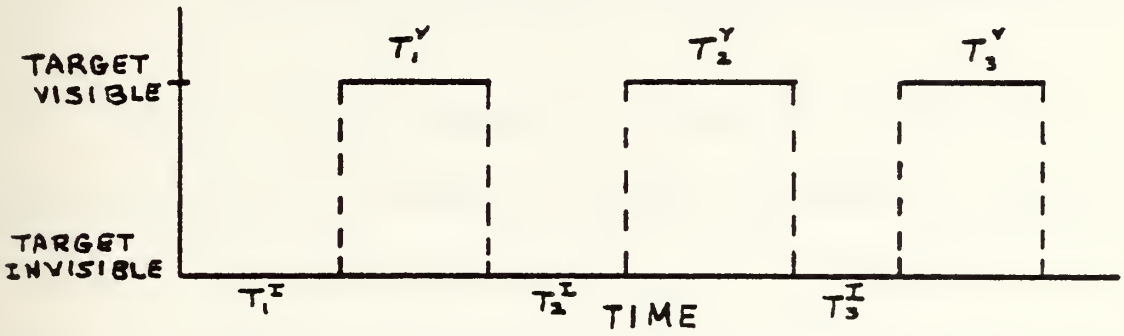


Figure 8: Two-state Stochastic Graph of Intervisibility

Since this is a stochastic process, there is an exponential distribution associated with each of the two states. This distribution represents the time between two occurrences of the same state. On the graph in Figure 8, the time between the end of T_1^V and the start of T_2^V is exponentially distributed with mean, η. Likewise, the time between the end of T_1^I and the start of T_2^I is exponentially distributed with mean, μ. Graphically, this can be represented by Figure 9, where the circles represent the two states, visibility and invisibility, and the arrows represent the transition from one state to the other at the given rate.

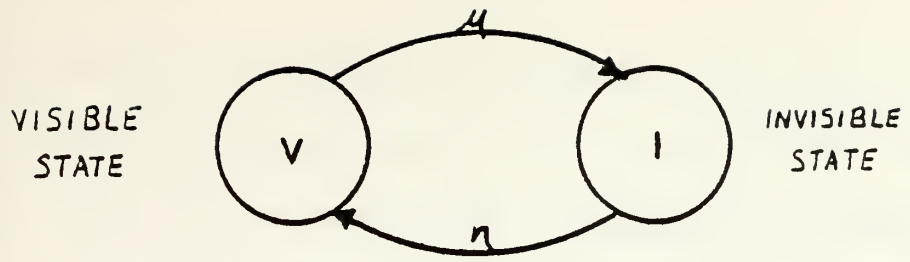


Figure 9: Two-state Stochastic Model of Intervisibility

The attrition caused by a firer on a target can occur only when the target is visible (assuming no attrition from wild, blind shots or indirect fire). Therefore, the attrition rate is a function of the probability that the target is visible to the firer. This probability will be denoted as P_v, and can be expressed as the proportion of time that the target is in the visible state in a steady state situation, $p_v = \frac{\eta}{\eta + \mu}$.

2. The Target Acquisition Process

Assuming now that line of sight exists between a firer and a target, the process of target acquisition must now be represented. There are two primary modes of target acquisition:

Serial acquisition, and

Parallel acquisition.

In the serial acquisition process, targets are acquired between engagements. That is, in concept, the firer is

concentrating on the engaged target and does not or is not able to acquire other targets on the battlefield. Any additional acquisitions made prior to engaging the current target are lost and must be re-acquired after the current engagement has ended. Examples of firers who use serial acquisition are:

any firer who is "buttoned up" and has his field of view narrowed to a thin band;

wire guided missiles/anti-tank systems.

In the parallel acquisition mode, the firer maintains his list of acquired targets throughout an engagement (except those lost through loss of line of sight). Thus, when an engagement has ended, by the target being killed or loss of line of sight, the firer may immediately engage a target that was previously acquired and is still a valid target. All systems which do not use serial acquisition are assumed to use the parallel mode.

For purposes of this thesis, all firers are assumed to use parallel acquisition. Therefore, serial acquisition will not be discussed any further. The reader should note that the serial acquisition process does have validity in modelling action on the battlefield, and should be considered in future enhancements of this model.

To explain the derivation of the target acquisition portion of the attrition rate equation, the diagram in Figure 9 must be modified to that in Figure 10.

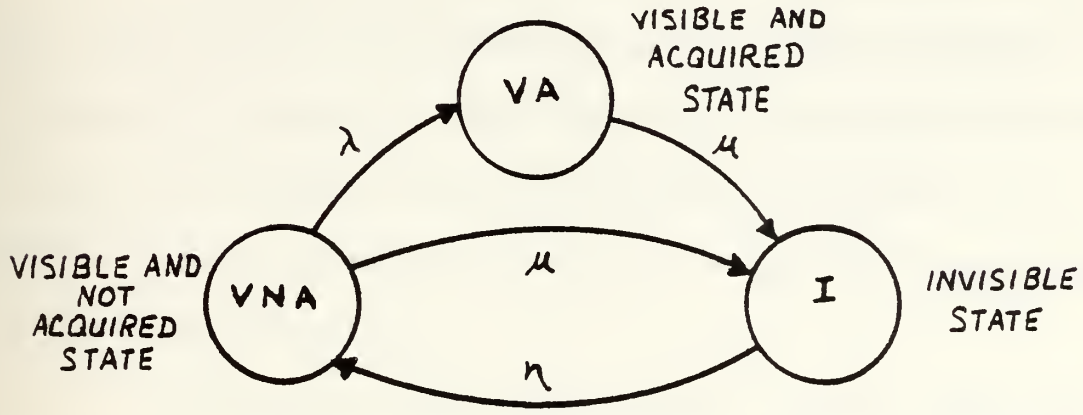


Figure 10: Three-state Stochastic Model of Intervisibility

The two-state stochastic process has been changed to a three-state process by dividing the visible state into the visible-not acquired state (VNA) and the visible-acquired state (VA). What has occurred is that a target can move from the visible-unacquired state to the visible acquired state at a rate λ. Since both of these states comprise the visibility state in total, the rate of moving from either of them to the invisibility state remains at μ. Notice that there is no direct transition from the invisible state to the visible-acquired state. This is only a matter of representation, since it is assumed that no more than one event, ie. moving to a new state, will occur during a time

increment. Thus, the situation described can still be represented by Figure 10.

Another point of interest in this diagram is the fact that there is no transition from visible-acquired to visible-not acquired. This implies the assumption that once a target is acquired, it remains acquired unless it is killed or line of sight is lost, in which case it transfers to the invisible state.

To determine the effect of the target acquisition process on the the attrition rate, the probability that a target is acquired must be determined. From Figure 10, and using the same concept of a stochastic process as that used in section III-D-1, the equation can be written

$$P_a(\text{tgt acquired}) = \frac{\lambda}{\lambda + \mu} . \qquad (1)$$

This should more properly be stated as the probability that a target is acquired given that it is visible, since this is actually a conditional probability with the state of acquisition being dependent on the target being visible.

Therefore, the probability that a target is visible and acquired in the steady state is:

$$\begin{aligned} P_{va} &= P(\text{tgt vis \& acq}) \\ &= P_a(\text{tgt acq} \mid \text{tgt vis})\, P_v(\text{tgt vis}) \qquad (2) \\ &= \left(\frac{\lambda}{\lambda + \mu}\right) \left(\frac{\eta}{\eta + \mu}\right) \end{aligned}$$

The implications of this equation will be discussed later in Chapter V.

3. Formulating the Attrition-rate Equation

Using equation (2) in the attrition-rate equation, the attrition rate of a single firer on a single target can be summarized using expected values as:

$$dx/dt = -P_{va}\,\alpha \qquad (3)$$

That is, the attrition rate equals the rate of killing an acquired target times the probability that a target is visible and acquired. However, since there are likely to be more than one firer and more than one target on the battlefield, this equation must be modified to reflect this.

If a firer is going to cause attrition on his targets, he must be able to acquire one or more of these targets. For firer y1 with x targets, the probability that y1 acquires one or more targets is just one minus the probability that he acquires no targets. Assuming the same probability of acquiring any of the targets, the probability of acquiring none of the targets is $(1 - P_{va})^x$. Therefore,

$$P_{a_{y1}} = 1 - (1 - P_{va})^x \; . \qquad (4)$$

The attrition rate now becomes

$$dx/dt = -P_{a_{xy}} \alpha \qquad (5)$$

for a single firer against x targets. Since there are more than one firer on the battlefield, each assumed to be acting independent in this process, the attrition rate in equation (5) must be multiplied by the number of firers(y) and the equation is now complete:

$$dx/dt = -P_{a_{xy}} \alpha \, y \, . \qquad (6)$$

In terms of the probability of acquiring a target, this equation can be written as:

$$dx/dt = -(1 - (1 - P_{va})^{x}) \, \alpha \, y \, . \qquad (7)$$

This is the same equation which Clark uses in his COMAN model, that uses results from a high resolution model to estimate the value for the conditional casualty rate and the probability that a target is not acquired. Henceforth, $(1 - P_{va})$ will be referred to as p for RED firer/BLUE target combinations (or q for a BLUE firer/RED target combination).

4. The Target Engagement Process (Bonder vs Clark)

The target engagement process can be modelled in its most detailed form, as in the high resolution model, or in

its simplest form, as in the aggregated, analytical model. The high resolution model, such as STAR, includes explicit representation of every factor that is involved in the process of aiming at, firing at, hitting, and killing a target. These factors include:

weapon system characteristics;

weapon/ammunition type characteristics;

target characteristics such as size, shape, vulnerability, etc.

In contrast, the low resolution, aggregated model represents the engagement process by one value--the attrition rate coefficient. This is the method to be employed in the model presented here. The task now is to develop a method to accurately estimate this coefficient for use in the attrition rate equation. Note that we are actually talking about two attrition rate coefficients and two equations. The derivation and method for estimating one is the same as the other, so for purposes of the discussion, references to an attrition rate coefficient will be in the singular.

There are presently two basic approaches to the problem of estimating attrition rate coefficients for parallel acquisition and stochastic line of sight. The first is

the use of the independent analytical model, involving weapon system characteristics and the use of algorithms to determine the expected time for a firer to kill a single target. The attrition coefficients are then the reciprocal of these expected times. This method will be referred to as Bonder's method after Seth Bonder who has used it extensively in his combat models such as VECTOR II.

The second method is the fitted-parameter analytical model, involving the use of a separate high resolution, Monte Carlo combat simulation to produce data to determine a maximum likelihood estimate of the attrition coefficients. This method will be referred to as Clark's method, since Gordon Clark popularized it in his doctoral thesis and in the development of the Combat Analysis Model (COMAN).

Most of this section involves a discussion of Clark's method and the practical application of it. However, in order to provide a contrast to that method, a short description of Bonder's method will be provided.

In general, the concept of Bonder's method is to produce an algorithm representing the expected time that it takes a firer to kill a single target. The elements of this algorithm include computations for line of sight, target acquisition, target selection, and the killing process.

Each element accounts for a certain amount of time in the expected time to kill a target. This entire process is seen by Bonder as a Markov renewal process where the outcome of a round fired is dependent only on the outcome of the previous round. By using weapon systems characteristics data as raw input, times to acquire, select, and kill a target can be determined and used in the algorithm for expected time to kill a target.

Bonder's method uses an independent, analytical model requiring a large data base of factors, such as range dependent weapon system characteristics, quantifications of line of sight and target acquisition parameters, to support the computation of the attrition rate coefficient.

A detailed explanation of Bonder's method for estimating the attrition rate coefficients can be found in the basic documentation for the VECTOR II model [Ref. 5]. Basically, Bonder says that the attrition rate coefficient A_{ij} is represented in the parallel acquisition mode by $A_{ij} = Q_{ij}\,\alpha$, where Q_{ij} is the probability that a given group-i weapon is firing at a group-j target. Q_{ij} can be expressed in terms of the probability that a group-j target is visible and has been detected by a group-i firer. This probabiltiy is denoted as S_{ij}, and the expression for Q_{ij} is :

$$Q_{ij} = S_{ij} \prod_{j=1}^{j-1} (1 - S_{ij}) \text{ for } j \geq 2 .$$

Assuming homogeneous forces, this equation reduces to

$Q_{ij} = S_{ij}$, and the i-j subscripts can be eliminated since there is no distinction among types of force elements. His resulting expression for S is identical to that of equation 4 , where S_{ij} and $P_{a_{xy}}$ are defined the same.

Bonder's expression for the conditional casualty rate, α , is written as the inverse of the expected time to kill an acquired target. This expression, when further broken down contains elements dependent on the weapon system characteristics, target characteristics, and a number of other subsystem performance parameters. These parameters were determined empirically outside the operational environment of the battlefield and molded together in this predetermined structure to produce the value of the attrition coefficient. This derivation of the attrition coefficient is the point where Bonder's method and Clark's method differ.

Clark's methodology is based on the assumption that the process of attrition in combat is a nonhomogeneous, two independent type Poisson Process where the time between casualties has an exponential distribution with parameter λ , called the total conditional casualty rate. Likewise, the

time between RED casualties and the time between BLUE casualties each has an exponential distribution with parameters λ_r and λ_b respectively, where $\lambda = \lambda_r + \lambda_b$. From the theory of Poisson Processes, the probability of a RED casualty occurring is λ_r/λ, and the probability of a BLUE casualty occurring is λ_b/λ. With this basis, Clark is able to develop a likelihood functon in terms of the conditional casualty rates α and β, of which the casualty rates are functions, ie., $\lambda_b = f(\alpha)$ and $\lambda_r = f(\beta)$. [Ref. 6] It should be noted that, in order to maintain consistency of terminology, the equations attributable to Clark's work have been adjusted to reflect the definitions presented in this thesis.

Clark also uses the concept that attrition is a function of target acquisition, which depends on the probability of a firer acquiring a target. This concept produces two more parameters which must be estimated: p, the probability that a BLUE target is not acquired by a RED firer; q, the probability that a RED target is not acquired by a BLUE firer.

For the functional form in Clark's COMAN model, the equations can be written as:

$$f_b = -\alpha[1 - p^m] \quad \text{with } m(0) = m_o$$
$$f_r = -\beta[1 - q^n] \quad \text{with } n(0) = n_o \qquad (8)$$

where the parameters are defined by:

f_b = the conditional attrition rate function for BLUE forces

f_r = the conditional attrition rate function for RED forces

α = conditional kill rate for a RED weapon against acquired BLUE targets

β = BLUE weapon conditional kill rate corresponding to α

p = probability that a BLUE target is unacquired by an individual RED firer

q = probability that a RED target is unacquired by an individual BLUE firer

m = number of BLUE firers surviving

n = number of RED firers surviving

These functional forms depend directly on the ability of firers to detect targets. To see this, consider just the conditional attrition rate function for the BLUE force, f_b. The definition of p implies that the value of $(1-p^m)$ is the probability that a RED firer has at least one acquired

BLUE target. Therefore, the attrition rate for the BLUE force is the kill rate for acquired targets times the probability of having an acquired target times the number of firers.

It should be noted that Lanchester's classical attrition rate functional forms are special cases of Clark's attrition rate functional forms, and, under appropriate conditions, Clark's equations reduce to Lanchester's equations. When p and q are zero, or very nearly zero,

$$f_b = - \alpha (1-0^m) n = - \alpha n \qquad (9)$$

and,

$$f_r = - \beta (1-0^n) m = - \beta m \qquad (10)$$

which are Lanchester's classical aimed fire equations. When individual targets become increasingly difficult to acquire, p and q assume values close to one and it can be proven that in the limit:

$$f_b = - \alpha (1-p) mn \qquad (11)$$

and,

$$f_r = - \beta (1-q) mn \qquad (12)$$

If the attrition rate coefficient a is set equal to $\alpha(1 - p)$ and the coefficient b is set equal to $\beta(1 - q)$, then the two equations above are identical to Lanchester's equations for area fire.

In his doctoral dissertation, Clark developed the likelihood function for a heterogeneous model, however since the DAMSTAC model assumes homogeneous forces, the necessary equations have been reduced to a simpler form than that derived by Clark. By differentiating the logarithm of the likelihood function with respect to p, the following equation is derived:

$$\frac{\partial \ln L(p,\alpha)}{\partial p} = -\sum_{k=1}^{K} \frac{C_k m_{k-1} p^{m_{k-1}-1}}{(1 - p^{m_{k-1}})} + \sum_{k=1}^{K} m_{k-1} p^{m_{k-1}-1} n_{k-1}(t_k - t_{k-1}) = 0 \qquad (13)$$

where

$C_k = 0$ if casualty k is RED

1 if casualty k is BLUE

m_{k-1} = number of BLUE survivors prior to kth casualty

n_{k-1} = number of RED survivors prior to kth casualty

$t_k - t_{k-1}$ = time between casualty k & casualty k-1

Likewise, by differentiating the logarithm of the likelihood function with respect to α, and by solving for $\hat{\alpha}$, a second equation in terms of $\hat{p}$ and $\hat{\alpha}$ is produced:

$$\hat{\alpha} = \frac{\sum_{k=1}^{K} C_k}{\sum (1 - \hat{p}^{m_{k-1}-1}) n_{k-1} (t_k - t_{k-1})} \qquad (14)$$

Differentiating with respect to q and β produces equivalent type equations which will allow the estimates for $\hat{q}$ and $\hat{\beta}$ to be determined in the same manner as follows.

Generally, the two equations (13) and (14) can now be solved simultaneously to determine the values for $\hat{p}$ and $\hat{\alpha}$. Unfortunately, this is easier said than done. The complexity of these equations (more so in the heterogeneous case) does not allow explicit equations for $\hat{p}$ and $\hat{\alpha}$. The remainder of this chapter will discuss a method for determining the estimates of p and α from these equations and a way to extract the necessary data from a high resolution Monte Carlo simulation run to compute these estimates.

E. ESTIMATING PARAMETERS FROM A HIGH RESOLUTION MODEL

As mentioned before, equations (13) and (14) must be solved simultaneously to produce values for $\hat{\alpha}$ and $\hat{p}$. Generally, these equations are too complicated to produce such values easily so other techniques must be used. The method suggested by Clark is to enter a value for $\hat{p}$ in equation (14) and solve for $\hat{\alpha}$. This value and the selected value for $\hat{p}$ are then put in equation (13) and the equation is evaluated. Notice that both values are estimates since we are computing the maximum likelihood estimates of p and α. The values of $\hat{p}$ and $\hat{\alpha}$ which give a value for equation (13) closest to zero are the best maximum likelihood estimates for p and α. If equation (13) equalled zero exactly, the true values of the maximum likelihood estimates would be known.

In using these equations, the input received from the high resolution, Monte Carlo simulation must include:

the time between casualties;

the number of RED and BLUE survivors prior to the kth casualty;

the total number of casualties for RED and BLUE for the battle;

the type of casualty (RED or BLUE) of the kth casualty.

In Clark's theory for a homogeneous force battle, the conditional attrition rates and the probabilities that a target is not acquired are assumed to be constant over the entire battle. This means that equations (13) and (14) are summed over all casualties in the battle and the estimates for p and α, which are determined from them, are also assumed to be constant over the entire battle.

In applying this theory to STAR output, the authors obtained questionable results for the values of the parameter estimates. The estimates, $\hat{p}$ and $\hat{q}$, were computed to have values of 0.99 and 0.93 respectively. These values imply that both BLUE and RED forces experienced difficulty in acquiring targets. Analysis of the Fulda Gap terrain and consideration of the respective locations of the opposing forces support the estimated values for p and q. However, the stated value for $\hat{p}$, 0.99, is a truncation of the actual value computed. If proper rounding procedures had been used, $\hat{p}$ would have been set equal to 1.0, which means that the RED forces would not detect any BLUE targets. Nondetection implies that no BLUE attrition will occur. But this contradicts the STAR battle which did, in fact, produce BLUE casualties amounting to about two-thirds of the total BLUE force.

The estimates, $\hat{\alpha}$ and $\hat{\beta}$, were computed to be 2.4 and 0.03, respectively. The relatively large value of $\hat{\alpha}$ implies that when the RED forces are able to detect BLUE targets, the BLUE force will suffer heavy casualties. There appears to be an unusually large difference between the values for $\hat{\alpha}$ and $\hat{\beta}$ which are not reflected by the values for $\hat{p}$ and $\hat{q}$. This disparity is most likely attributable to the restriction that all parameter values be held constant over the entire battle. Experimentation with different parameter values in DAMSTAC indicates that a value of about 0.40 more closely approximates the attrition observed in the STAR battle. Experience with these computations reveals that this is not necessarily the case and that a more likely assumption is that the parameters are constant over different time intervals of the battle. In general, a simple battle can be divided into phases where the intensity of the battle is constant within each phase but different among the phases. Basically, the phases are:

movement to contact (low intensity)

initial contact (mid intensity)

main battle (high intensity)

withdrawal from contact or annihilation (mid/low intensity)

Moving from one phase of the battle to the next should cause a change in conditional casualty rates (α and β) as well as the probability of not acquiring a target (p and q).

The question now becomes how to determine the breakdown of the battle by phases using the output from a high resolution, Monte Carlo simulation. That is, where do the changes in phases occur in the simulation? One possible method is to look at the number of casualties as a function of time. Another is to look at the number of shots fired as a function of time (see Figure 11). In both cases, time is segmented into time intervals (eg. every 100 seconds) and a histogram of the casualties or shots is plotted for each of these intervals. Having done this for several battles, the results identify fairly clear choices for the end of one phase of the battle and the beginning of the next phase. However, more analysis of these two methods reveals that they might not truly identify the phases of the battle and that an alternate method should be considered, primarily one based on range. This concept is discussed in detail in Chapter VI.

Using the number of casualties per time interval, a distorted picture of the intensity of the battle might be obtained. The interaction of the probability of acquiring a

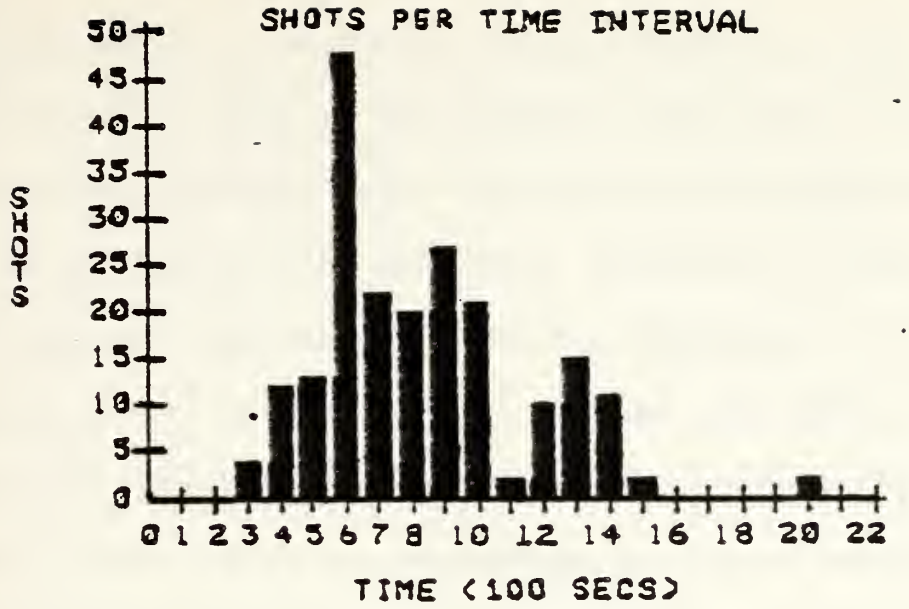


Figure 11: Histogram of Shots per Time Interval

target with the attrition rate of a firer might offset each other and produce results which conflict with the true battle intensity. In other words, if the acquisition probability is high but the attrition rate is low, the number of casualties in a time interval might indicate a low intensity phase of the battle when, in fact, it is a high intensity phase. Therefore, this method should be used with caution, if at all. The method of using the number of shots per time interval has similar faults. Twenty shots by one firer in a time interval is much different in terms of intensity than one shot by twenty firers. Again, this method should be used with caution.

The question is then: what method should be used? One method, which is used by the Combined Arms Combat Development Activity (CACDA/CASAA), is measuring the number of shots per firer per time interval.[1] Although it also has some faults, it does provide a better indication of the intensity of the battle. It is, for the most part, independent of the conditional casualty rates, although the number of shots would indicate a dependency on target acquisition. This dependency is reduced by averaging the number of shots over the number of firers. In effect, this measure can be considered to be unbiased by the changes in conditional casualty rates and probabilities of not acquiring targets. Thus one is able to get a feel for the intensity of the battle from the histogram of the number of shots per firer per time interval. This is not necessarily scientific reasoning, but it seems to work.

Once the phases of the battle are delineated for a high resolution, Monte Carlo simulation run, the next step is to determine how to treat the 'dead' time from the beginning of a phase to the first casualty and from the time of the last casualty to the end of the phase. For phases in the middle

---

[1]Conversation with Mr. Ernest Boehner from CASAA on 5 December 1981

of the battle, each of these two times is a portion of the time between two casualties and should thus have a bearing on our maximum likelihood estimates from equations (13) and (14).

To resolve this situation, the time of the last casualty within a phase is used as the end of the phase. For example, if a phase is supposed to end at 1200 seconds and casualties occur at 1140 and 1225 seconds, the phase will be terminated at 1140 and the new phase begun at 1141. This also allows this time between casualties to be accounted for in the next phase. This method produces some variation in phase lengths when running several replications of the battle, but this variation is minimal, especially since the beginning and end of phases are being judged from something less than a scientific method.

Several replications, as many as affordable, should be used to compute the maximum likelihood estimates for the four parameters. By using a large sample population (the casualties), a more accurate estimate of the parameter can be obtained. Likewise, when the heterogeneous model is used, a larger sample population is desirable, since the number of engagements between two particular weapons systems will be much less than the number of engagements in the

homogeneous case. This could produce values that are statistically insignificant.

A postprocessor to the high resolution model was developed to make the numerous calculations required to produce these estimators. The postprocessor uses an optimization routine to determine the maximum likelihood estimates from the given equations. Detailed information on the postprocessor is provided in Appendix E.

IV. ALTERNATE FUNCTIONAL FORMS

Different military situations have been hypothesized to yield different functional forms for Lanchester-type equations. The term "Lanchester-type equations" refers to any differential-equation model of combat and the term "functional form" relates to the structure of the attrition rate, i.e., how the attrition rate equation is written and which parameters are considered explicitly in the equation. Henceforth, a standard shorthand notation will be adopted to refer to the functional forms used in homogeneous-force Lanchester-type combat models. For example, Lanchester's equations of modern warfare are expressed as F|F functional forms (or, simply F|F attrition) because they are proportional to only the number of enemy firers. Taylor [Ref. 7] provides a good explanation of this convenient shorthand notation.

Generally, the classical Lanchester functional forms for expressing the attrition rate for forces using aimed fire or area fire are too simple to be used "as is" in a detailed analytical combat model. The assumptions required for these basic equations to hold are too limiting and do not allow for all of the processes that are known to occur in combat

to be considered in the force attrition model. Some of the more common functional forms for attrition rates that have been considered and used in Lanchester-combat models are shown in Table 2.

Of special interest is the T|T functional form. This functional form is particularly useful in describing attrition in target rich environments. Peterson [Ref. 8] hypothesized that the equations,

$$dx/dt = -ax \text{ and } dy/dt = -by,$$

characterize the early stages of a small unit engagement in which the vulnerability of a force dominates its ability to acquire enemy targets. It is recommended that Peterson's T|T functional form be considered in the heterogeneous model described in Chapter VI to model attrition in the initial phase of the battle. The remaining sections of this chapter will discuss several not-so-common functional forms that have been formulated and used in "recently" developed combat models.

A. HELMBOLD-TYPE EQUATIONS

An alternative functional form for homogeneous-force attrition rates was proposed by R. Helmbold. Helmbold believed that when opposing force sizes were grossly unequal, the larger force experienced "inefficiencies of scale"

Table 2: FUNCTIONAL FORM NOTATION

FUNCTIONAL FORM	DIFFERENTIAL EQUATION
F|F	$dx/dt = -ay$ $dy/dt = -bx$
FT|FT	$dx/dt = -axy$ $dy/dt = -bxy$
F|FT	$dx/dt = -ay$ $dy/dt = -bxy$
T|T	$dx/dt = -ax$ $dy/dt = -by$
(F+T)|(F+T)	$dx/dt = -ay - cx$ $dy/dt = -bx + dy$

in producing casualties. Limitations on space available for maneuver, reduced reaction time, increased command, control, and communication problems, etc. result in the reduction of casualty producing effectiveness. These limitations may well prevent the larger force from using its full destructive capability against the smaller force. There is no definition for "grossly" unequal force sizes, but the following hypothesis is proposed. The validity of using Helmbold's equations in a combat model is directly related not only to the force ratios (i.e., x/y or y/x) but to the relative size of the twc cpposing forces with respect to their surrounding terrain. For example, a 5 to 1 force ratio may

well be a situation where Helmbold's equations apply if both of the opposing force sizes are relatively large with respect to the terrain available to them for movement, concealment, etc. Conversely, a force with a 30 to 1 force ratio may not exhibit inefficiences of scale if the opposing force sizes are small in relative sizes and with respect to their surrounding terrain. Consequently the decision to use Helmbold's functional form for attrition rates rather than some other form (like Lanchester's equation for modern combat) really depends on the tactical "situation".

Helmbold's functional form introduces a modification that accounts for the reduced fire effectiveness of the larger force. In mathematical form his equations would read:

$$dx/dt = -a\, g(x/y)\, y \qquad \text{with } x(0) = x_0$$

$$dy/dt = -b\, h(y/x)\, x \qquad \text{with } y(0) = y_0$$

where a and b represent time-dependent attrition-rate coefficients, and $g(x/y)$ and $h(y/x)$ represent functions that modify fire effectiveness of individual x and y firers respectively. According to Helmbold's hypothesis the two functions $g(x/y)$ and $h(y/x)$ must satisfy:

$g(1) = h(1) = 1$

$g(u) = h(u) = E(u)$

$E(u)$ is a strictly increasing function of u.

So, the generalized Helmbold-type combat equations can be written as:

$$dx/dt = -a\ E(x/y)\ y$$

$$dy/dt = -b\ E(y/x)\ x$$

For the case in which $E(u) = u^c$ with c a constant greater than or equal to zero, the following equations for Helmbold-type combat are obtained:

$$dx/dt = -a\ (x/y)^{1-W}\ y \quad \text{with } x(0) = x_o$$

$$dy/dt = -b\ (y/x)^{1-W}\ x \quad \text{with } y(0) = y_o$$

W is commonly called the "Weiss parameter" with W = 1-c.

The above equations represent the special exponential case of Helmbold combat. If a(t) and b(t) represent constant attrition rate coefficients denoted by a and b respectively, then, when W = 1, the above equations reduce to Lanchester's Equations for Modern Combat. When W = 1/2 Lanchester's Area-fire Equations are obtained.

B. HOW TO CHOOSE THE RIGHT FUNCTIONAL FORM

A very logical question that readers of this study may ask at this time is, "Now that I've seen some of the Lanchester-type attrition rate functional forms, which one should I choose for my analytical model?" The answer to this question is that there is no answer to this question. The choice of functional form is purely arbitrary and is

usually left to the discretion of the model builder. The following steps, however, may help in this choice.

1. Carefully consider the scenario that the combat model is intended to portray. The scenario may well indicate that some particular subset of the functional forms is preferable.
2. List the assumptions that must be made for each of the functional forms in the subset to hold and see which assumptions are acceptable to you and your model and which are not.
3. Observe the parameters involved in each functional form and eliminate those forms for which the parameter values or estimates are unobtainable or for which you are not willing to plug in arbitrary values.

At this point, the final selection of functional form is purely one of subjective judgement.

There is no requirement, however, that a model builder must accept one of the already developed functional forms. There is no doubt that other forms can be formulated, but the formulation will require extensive research, analysis and work. A good check to see if an attrition rate equation is reasonable is to insure that the Lanchester classical equations are special cases of the new functional form.

Once the final functional form has been selected or formulated, it may be enhanced as necessary to reflect more of the complexity and realism of combat. Some additional operational factors that can be considered are:

Unit breakpoints and battle termination criteria;

Suppression;

C^3;

Reinforcements or replacements.

The main assumption that allows these operational factors to be "added" to the attrition rate equation is that synergistic effects are not considered in Lanchester-type force-on-force combat models. No synergistic effects imply that the Principle of Superposition holds, therefore, the effects of the operational factors are independent and additive.

V. FUTURE ENHANCEMENTS

There are numerous enhancements to the analytical model that can be made. The enhancements that are recommended, for the most part, are not aimed at improving the inner workings of the current model, but at building successive models that are more detailed, require fewer limiting assumptions, and are generally more complicated.

For follow-on models that retain the homogeneous force concept, it is recommended that the model have the ability to allow the user to choose alternative functional forms for the force attrition rates. This should be done to better describe the situation of the battle at a particular time. A likely method would be to change from one functional form to another during the battle in order to more realistically portray the attrition occurring then. This enhancement would require considerable effort in developing the maximum likelihood estimates for the various parameters used in each alternative functional form, additional work with the STAR postprocessor, and significant change to the current DAMSTAC model. Questions which arise from this suggestion are:

Which form(s) should be chosen?

How do you know when to change functional form?

These questions are partially answered in Section IV-C. Further research into this area is required.

The movement subroutine of the model needs to be modified to more realistically portray the movement of units on the battlefield. Even though a relatively low-resolution time-step operation is occurring, the resolution of the movement of units can be refined to vary the rate of movement from one time step to the next, if necessary. Currently, a change in rate of movement occurs only at the initial contact between forces and at breakpoints. The rate of advance of the RED forces should be allowed to vary based on decision logic which considers attrition rate, terrain, etc. BLUE forces should be given the capability to move to alternate or fall back positions when the battle situation dictates. Decision logic must be enhanced to allow BLUE forces to move to alternate or fall-back positions. It should be noted that this "pace of battle" question has never been fully explained or understood, however, the movement routine logic in DAMSTAC should parallel that of STAR.

The concept of units having different formations is ignored in the current model. Units are identified by a

point location only. Some work needs to be done to allow the units to take on different formations for road marches, attacks, etc. In order to do this, additional attributes such as frontage, depth, orientation and geometric configuration should be considered to describe the formation of the unit. A postprocessor for STAR could take the individual tank locations from the STAR data base and determine the above mentioned attributes for input into the DAMSTAC model. Attrition rates on both sides must be adjusted for the various formations of an attacking unit.

The inclusion of indirect fire, suppression, smoke, intelligence, and close air support are all important to making the model a usable analytical tool. Additions of communications and logistics, primarily ammunition and petroleum, oil, and lubricants (POL), are also important enhancements that must be included in the model.

As was seen in Section III-E-3, P_{va}, the probability that a target is visible and acquired in the steady state is

$$P_{va} = P_a \; P_v = \left(\frac{\lambda}{\lambda + \mu}\right) \left(\frac{\eta}{\eta + \mu}\right) \qquad (15)$$

Both of the factors represent a particular element of the attrition process, one the terrain affects on line of sight,

and the other, the acquisition process. It should be apparent that these two factors can be separated from each other with the only common variable, μ, the rate of transfer from the visible state to the invisible state. With this factorization, if the line of sight parameters for a particular terrain type are known and the attrition rate coefficients have been determined, the target acquisition capability of a weapon system against another weapon system can be determined. This would allow the target acquisition parameter to be used for other terrain types to compute attrition rate coefficients without having to estimate the coefficients from a high resolution simulation.

The concept works in this manner. Runs are made on the high resolution model (STAR) using a particular terrain. The attrition coefficients are determined using the maximum likelihood estimates described in Section III-D. With predetermined data for the terrain coefficients, the line of sight parameter can be factored out of the attrition rate coefficient, A. This would leave the value of the acquisition factor which should theoretically be constant for the firer-target type combination used in the scenario. With this value and the terrain coefficient for other types of terrain, the attrition rate coefficient can be determined

for a new scenario with the same firer-target combination, without running the high resolution model again. This method assumes that the terrain affects and the target acquisition process are independent of each other. The impact of this concept is enormous in terms of time and cost saved from not having to make multiple runs of the high resolution model.

One concept which is not strictly an enhancement but is an important consideration is the idea of using simulated combat data from a source other than STAR. Specifically, data generated at the National Training Center from actual field exercises could conceivably be used to compute attrition coefficients in much the same manner as STAR. What this provides is not only an alternate source of data, but also a different model, since it would include many human factors that would not likely be accounted for in STAR. Depending on the intended use of the model, one source or the other may be desirable.

It is necessary that a progression from homogeneous forces to heterogeneous forces be made if DAMSTAC is ever to be really useful. This enhancement entails a significant amount of research, model building, and statistical analysis in order to implement the concept of heterogeneous forces

into the analytical model. Target priorities, fire allocation algorithms, and a library of attrition rate coefficients must be developed for various scenarios and force mixes. Changing the model to a heterogeneous type requires some changes in the equations (13) and (14), and thus some significant modifications to the postprocessor for the model.

VI. HETEROGENEOUS MODEL

DAMSTAC, in its current configuration, models combat between two opposing homogeneous forces, but actual combat consists of many different weapon-system types operating together as combined arms teams. For example, there may be infantry, tanks, antitank weapons, artillery, etc. on each side. A necessary extension or modification, therefore, to DAMSTAC is the consideration of combat between heterogeneous forces. This chapter briefly indicates how the basic ideas and methodologies used previously to model combat attrition for homogeneous forces can be extended and adapted to the heterogeneous force structure.

A key assumption that must be made in order to model combat between heterogeneous forces, where Lanchester-type differential equations are used, is that the attrition effects of various different enemy weapon systems types against a particular friendly target type are additive. In other words, synergistic effects between weapon system types are not considered [Ref. 9]. Also inherent in the modelling of heterogeneous force combat is the need to introduce an allocation factor. This allocation factor may be defined as the fraction of the firers of a particular type that engage

a particular target type. The allocation factor concept will be discussed in greater detail later in this chapter.

One theoretical concept for modelling the attrition process between two opposing forces is to consider attrition as a nonstationary Markov process where the states of the process are defined as the numbers and types of surviving combatants and the ranges between opposing combatants. The transition probabilities between the Markov states, under this concept, depend on the ratio at which surviving combatants or weapon systems inflict casualties on a particular type of opposing combatant. Thus the nonstationary Markov process model provides a means of relating the sequence of states and the times of the state transitions observed during a battle to attrition rate coefficients. In turn, this allows for the estimation of attrition rate coefficients on the basis of the observed sequence of states and transition times. Andrighetti [Ref. 10] developed such a nonstationary Markov model that related weapon system effectiveness to the time sequence of casualties observed in a two-sided, heterogeneous force land combat simulation. There is a flaw, however, in Andrighetti's work that readers should note. He implies that his methodology is easily extended to time-dependent attrition rate coefficients. This is not true.

Andrighetti's thesis does contain a good discussion on the theoretical background which allows for the maximum likelihood estimation of parameters used in attrition rate equations. An important observation made by Andrighetti in his thesis is that the difference-differential equations obtained from the Chapman-Kolmogorov equations of the nonstationary Markov model are susceptible to a general solution only in special cases as Gordon Clark showed in his doctoral dissertation. Clark used the Chapman-Kolmogorov equations to develop expressions for time state probabilities and the expected survivors in a heterogeneous force battle.

The extended version of the DAMSTAC model which shall be called DAMSTAC II should be a heterogeneous force analytical model that uses STAR output to compute parameter values. These parameter values should be allowed to change as the distance or range between opposing forces vary during the battle. In DAMSTAC, an attempt was made to allow the parameter values to change with respect to time. However, it was discovered that the maximum liklihood estimators developed for the critical parameters in the attrition rate equations could not be used for time dependent variables. Two feasible alternatives that allowed for variable parameter values

were considered. Both alternatives are recommended for future evaluation. The first alternative looks at dividing the battle into phases based on battle intensity as measured by the number of shots divided by the number of live firers. (Refer to Section III-E) The second alternative for dividing the battle into phases is based on the range between combatants. Here also the parameters can change from phase to phase. This technique will work assuming that the combatants on each side move as a single force and that units are located by their centers of mass. For example, a battle can be divided into three distinct time intervals or phases of combat: an artillery prep phase during which all combat vehicles and weapon systems, other than artillery, are not participating; a moderate to long range battle phase; and, a terminal shorter range battle phase. Parameter values can then be computed, using procedures to be discussed later, for each battle phase. This second alternative is used in the COMANEW model and is the one recommended for DAMSTAC II.

A. THE ATTRITION-RATE FUNCTIONAL FORM

The attrition rate functional form that will ultimately be chosen for the heterogeneous force model is, of course, the prerogative of the modeller. The authors of this thesis, however, would like to recommend two related attrition

rate functional forms for consideration. The first, which shall be labeled as Functional Form 1 (FF1), is based on the assumption that during the course of a battle fire is directed at the highest priority target surviving and acquired. Also it is assumed that the relative priority of target weapon types, as considered by each firer, remains constant throughout the engagement and that this relative priority is identical for each firing weapon type. Functional Form 1, developed by Gordon Clark for use in the COMAN model, can be written as follows:

$$f_{bj}^{(i)}(m\,n) = \sum_{k=1}^{r} \alpha_{ijk}\, n_k\, p_i^{m_j'}\left(1 - p^{m_j}\right)$$

$$f_{rk}^{(i)}(m\,n) = \sum_{j=1}^{b} \beta_{ikj}\, m_j\, q_i^{n_k'}\left(1 - q^{n_k}\right) \qquad (16)$$

where

$f_{bj}^{(i)}(m,n)$ = conditional casualty rate of BLUE weapons of type j during phase i of the battle given strengths of m and n.

$f_{rk}^{(i)}(m,n)$ = analog of f_{bj} for RED weapons of type k.

n_k = the number of surviving RED weapons of type k

m_j = the number of surviving BLUE weapons of type j

β_{ikj} = kill rate of a BLUE firer of type j for an acquired RED target of type k in the i'th time interval.

α_{ijk} = kill rate for a red-firer/blue-target combination corresponding to β_{ikj}

q_i = the probability that a specific RED weapon is unacquired by an individual BLUE firer in the i'th phase of the battle.

p_i = the probability for a specified blue-target/red-firer combination corresponding to q_i.

m_j' = the number of surviving BLUE weapons of higher priority than a type j weapon.

n'_k = the number of surviving RED weapons of higher priority than a type k weapon.

The reader should note that since the assumption is made that a firer will engage the highest priority target he can acquire, the probability of a BLUE firer engaging a RED weapon of type k is $q_i^{n'_k}(1-q_i^{n_k})$. Similarly, the probability of a RED firer engaging a BLUE weapon of type j is $p_i^{m'_j}(1-p_i^{m_j})$. These two probabilities may be regarded as the allocation factors for Functional Form 1.

All of the critical parameters necessary to compute the conditional casualty rates, $f_{bj}^{(i)}(m,n)$ and $f_{rk}^{(i)}(m,n)$, can be obtained either directly or indirectly (using maximum likelihood procedures) from the output of a high resolution simulation such as STAR.

If one is not willing to assume that target priorities remain constant throughout the battle then FF1 will not be sufficient. The terms in Clark's functional form which result from the assumption of constant target priorities are $q_i^{n'_k}$ and $p_i^{m'_j}$ where $q_i^{n'_k}$ is the probability that all surviving RED weapons of higher priority than type k have not been acquired and $p_i^{m'_j}$ is the corresponding probability for type j blue weapons.

One way to extend Clark's formulation in order that the fixed priority may be dynamically overridden is by replacing the terms $q_i^{n'_k}$ with Θ_{jk} where Θ_{jk} is defined as the probability that BLUE weapon type j engages RED weapon type k, given that BLUE weapon type j acquires at least one RED weapon type k. Note that

$$\Theta_{jk}(1-q^{n_k}) = P(j \text{ engages } k \mid \text{at least one } k \text{ acquisition})$$

$$P(\text{acquire at least one } k) = P(j \text{ engages } k)$$

Functional Form 2 (FF2) may be written as;

$$f_{bj}^{(i)}(m,n) = \sum_{k=1}^{r} \alpha_{ijk}\, n_k\, \delta_{kj}\, (1-p_i^{m_j}) \tag{17}$$

$$f_{rk}^{(i)}(m,n) = \sum_{j=1}^{b} \beta_{ikj}\, m_j\, \Theta_{jk}\, (1-q_i^{n_k}) \tag{18}$$

where δ_{kj} is the RED firer BLUE target probability corresponding to Θ_{jk} and all other parameters are as defined for FF1. Just as p, q, α, β are assumed constant over the entire phase of the battle, δ and Θ, for each j and k, are also assumed constant over the entire interval.

Serious consideration should be given to incorporating FF2 or some similar functional form into DAMSTAC II because STAR does allow target priorities to change during a battle. Target priorities in STAR are based on the types of targets on the firer's detected list and the type of ammunition the firer has available for use.

It may also be useful to use a different functional form during the initial phase of combat where attrition rates are more a function of the exposure or vulnerability of targets than they are a function of the firer's capability to acquire targets. Clark recommended the use of Peterson's Logarithmic Law to model attrition during the initial phase of the battle. [Ref. 11]

B. MAXIMUM LIKELIHOOD ESTIMATORS

The ability of the DAMSTAC II model to provide insight into the interactions represented in the STAR simulation is based on the estimation of conditional casualty rates (α and β) and other parameters from STAR data. These estimated parameters should be constant within a phase of the battle, but independent and unrelated to results in neighboring time intervals. The independence of parameter values between time intervals implies that the estimators for a specific time interval, or phase of the battle, can be defined by

analyzing a sample of data from the high resolution simulation observed during that single-time interval.

The maximum likelihood estimators (MLE's) for the unknown parameters α, β, p and q used in FF1 were developed by Gordon Clark. It may be of interest to the reader to note that simplified versions of these MLE's were used in the DAMSTAC model for homogeneous forces, however, since we are now concerned with heterogeneous forces, the "unabridged" versions are presented. The maximum likelihood estimate for α as a function of p is:

$$\hat{\alpha}_{kj}(p) = \begin{cases} \dfrac{B_{kj}}{\sum_{y=1}^{y+1} n_{yk}\, p^{m'_{yj}}\,(1 - p^{m_{yj}})(t_y - t_{y-1})} & \text{for } B_{kj} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (19)$$

where $\hat{\alpha}_{kj}(p)$= the estimator for the parameter α for a given p.

The MLE for p is found by solving:

$$\frac{\partial \ln L(p)}{\partial p} = \sum_{y=1}^{y+1} \frac{(1 - C_y)(m'_{yg_y} - (m'_{yg_y} + m_{yg_y})p^{m_{yg_y}}}{p(1 - p^{m_{yg_y}})} - \sum_{y=1}^{y+1}\sum_{j=1}^{k}\sum_{k=1}^{r} \left(\hat{\alpha}_{jk}(p) n_{yk}\, p^{m'_{yj}-1}\,(t_y - t_{y-1}) \cdot (m'_{yj} - (m'_{yj} + m_{yj})p^{m_{yj}}) \right) = 0 \qquad (20)$$

The equations for $\hat{\beta}_{kj}$ (q) and $\hat{q}$ are similar.

The estimators obtained from the above equations result from the data from a single replication of the combat simulation. More precise estimates are obtainable by using a larger sample consisting of a number of observations during

the interval in question from a number of independent replications of the STAR simulation. This modification entails an additional summation ($\sum_{x}$, where x = the total number of replications) in each of the estimating equations. The data needed for the computations of the MLE's include;

y = total number of casualties

t_y = elapsed time since the beginning of the interval until the y'th casualty

t_{y+1}= the end of the battle phase or the elapsed time since the beginning of the phase until the battle ended if it is less.

m_{yj} = the number of BLUE survivors of type j just prior to the y'th casualty

n_{yk} = the number of RED survivors of type k in the battle just prior to the y'th casualty

C_y = 1 if the y'th casualty is BLUE ; 0 otherwise

f_y = firer weapon type for the y'th casualty

g_y = target weapon type for the y'th casualty

m'_{yj} = the number of BLUE survivors of higher priority than type j just prior to the y'th casualty

$m'_{yj} = \sum_{u=1}^{b} M_{yu}$

n'_{yk} = RED analog of m'_{yj}

B_{kj} = the number of BLUE weapon-type j casualties due to RED weapons of type k

One technique of computing the MLE's for $\hat{\alpha}$ and $\hat{p}$ is to numerically determine $\hat{p}$ using equation (20) and then substitute this value into equation (19) for j = 1,2,...,b and for k = 1,2,...,r (b = the total number of BLUE weapon types and r = the total number of RED weapon types). A recommended technique for numerically solving for $\hat{p}$, the estimate for p, can be found in Clark's work on the Dyntacs combat model [Ref. 12]. This technique is similar to the one developed for the homogeneous case discussed in Chapter III. Basically the procedure is as follows. Since p represents a probability, it must satisfy the inequality $0 \le p \le 1$. Therefore, the function $\frac{\partial}{\partial p}$ Ln L(p) can be evaluated at points in the interval $0 \le p \le 1$ until two points, p_1 and p_2, are found such that the product of $\frac{\partial}{\partial p}$ Ln $L(p_1)$ and $\frac{\partial}{\partial p}$ Ln $L(p_2)$ is less than or equal to zero. Since $\frac{\partial}{\partial p}$ Ln L(p) is a continuous function, $\frac{\partial}{\partial p}$ Ln L(p) = 0 for some p satisfying $p_1 \le p \le p_2$. This interval is reduced by evaluating $\frac{\partial}{\partial p}$ Ln L(p) at the midpoint of the interval (p_1, p_2). Repeating this process converges to a solution for $\frac{\partial}{\partial p}$ Ln L(p) and this solution is a local maximum. This technique is similar to the bisection method for solving nonlinear problems.

With respect to Functional Form 2, the introduction of the θ_{jk} allocation terms adds considerable complexity to the maximization of the likelihood function. One way to circumvent this problem is to compute a "best guess" for the θ_{jk}'s from STAR samples (for the given interval). This best guess can thenbe substituted into the θ_{jk}'s prior to maximization of the likelihood function, thereby reducing the θ_{jk}'s to constants. The maximization can then be performed, as in Clark's original equations, with respect to p, q, α, β .

C. THE MODEL

The computational procedures that will constitute DAMSTAC II must be developed by the modeller who chooses to carry on the work initiated in this thesis. As a starting point, however, the techniques used by Clark in his COMAN model would be excellent choices. The adequacy of Clark's procedures in DAMSTAC II given a particular scenario and force mix, to predict attrition in a STAR simulation with the same scenario and force mix is questionable and would have to be tested. Detailed comparisons and analysis of the output from both models will be required to answer the question of model comparability and modifications to the computer code and/or computational procedures in DAMSTAC II must be accomplished as necessary. The computational cycle

employed in COMAN is based on the assumption that the distribution times between casualties in the high resolution simulation is exponentially distributed. It uses Monte Carlo procedures to generate the time until the next casualty and also to find the casualty weapon type.

The following steps constitute the computational cycle recommended for DAMSTAC II.

Step 1. The first step in the cycle sets the time equal to zero and sets the battle phase, designated by the letter i, equal to 1.

Step 2. Next the total conditional casualty rate, $\lambda^i(m,n)$ is computed from the equation

$$\lambda^i(m,n) = \sum_{j=1}^{b} f_{bj}^{\dot{\omega}}(m,n) + \sum_{k=1}^{r} f_{rk}^{\dot{\omega}}(m,n) \quad \text{for } i = 1,2,\cdots, I \qquad (21)$$

where $f_{bj}^{\dot{\omega}}(m,n)$ and $f_{rk}^{\dot{\omega}}(m,n)$ are the conditional casualty rates in phase i of the battle, given force strength m and n, for BLUE weapon type j and RED weapon type k, respectively. $\lambda^i(m,n)$ is the parameter of the exponential distribution describing the times to the next casualty.

Step 3. Now, using the exponential distribution with parameter $\lambda^i(m,n)$, one Monte Carlo's for the time interval to the next casualty. Designate this time interval as t'. If the next casualty occurs in another time interval, ie. if $t + t' > t_i$ where t_i represents the end of the i'th battle

phase, then i is increased by 1, the time is set equal to t and the program is directed back to step 2 where $\lambda^{(i+1)}(m,n)$ is selected as the distribution parameter. If, however, $t + t' < t$; then the next step is to Monte Carlo for the casualty weapon type.

Step 4. The weapon type for the next casualty is represented by the random variables C_b and C_r which are defined as

C_b = j if the casualty weapon is BLUE weapon type j

0 if the casualty is RED

C_r = k if the casualty weapon is RED weapon type k

0 if the casualty is BLUE

The Monte Carlo procedure uses the conditional casualty weapon probability distributions for C_b and C_r. These probabilities are computed by

$$p^i(C_b = j|C = 1,m,n) = f^{(i)}_{bj}(m,n)/\lambda(m,n)$$

$$p^i(C_r = k|C = 1,m,n) = f^{(i)}_{rk}(m,n)/\lambda(m,n)$$

for i= 1,2,..., I and when variable C is set equal to 1 if a casualty is determined to occur in interval i from the previous steps. The probability that the casualty is of a particular type is computed for each RED and BLUE type weapon. The sum of these probabilities should equal 1.

Step 5. The next step is to Monte Carlo using the Uniform (0,1) distribution to determine the casualty type. An example may help in understanding the above procedure. Suppose there are two types of BLUE weapons and two types of RED weapons. Assume the conditional casualty probabilities for each weapon are computed using the equations shown previously and they are 0.25 for each weapon type, b1, b2, r1, r2. If one orders the weapon types in some arbitrary fashion such as b1, r1, b2, r2, and then accumulates the probabilities on the interval (0,1), one can assign an interval to each weapon type. The example would yield the following intervals:

b1 - (0,0.25);

r1 - (0.26,0.50);

b2 - (0.51,0.75);

r2 - (0.76,1.00).

The uniform random number can then specify the casualty weapon type. If the random number is 0.60 then the casualty weapon type in our example is b2.

Step 6. After the casualty weapon type has been found the next step is to decrement the number of that type weapon by 1. Now increase the value of t by t', the time to the next casualty. If t is greater than or equal to t_τ which

represents the ending time of the last phase of the battle, or if either side is annihilated , the battle has ended and the program terminates. If t is less than t_I then go to step 2.

Observe that the above computational procedures are stochastic in nature and different replications of an engagement will likely yield similar but different results. If this method is chosen for DAMSTAC II, variance reducing replications for each engagement should be performed.

D. THE STAR POSTPROCESSOR

It is very important that anyone who attempts to develop the DAMSTAC II model understand and be able to manipulate the STAR simulation model. In the initial attempts to extract data from STAR, certain subroutines must be turned off. These routines include the Smoke, Engineer, Close-air support, etc. whose effects on parameter estimates used in DAMSTAC II have not been thoroughly analyzed. The number of types of combatants played in the simulation will have to be regulated so that they do not overpower the capabilities of the analytical model. It is recommended that no more than three different types of weapon systems for each opposing force be attempted when initially formulating the heterogeneous force model.

It is important that the finished version of the analytical model be as simple and transparent as possible and not encumbered with parameter estimating algorithms and computations. This consideration plus the amount of data that must be extracted from the STAR output require the development of a STAR postprocessor program. As an illustration, if one considers a simple battle divided into three battle phases with only three types of weapon systems on each side, the DAMSTAC II model, as it has been described, requires the estimation of 114 parameter values and the computation of six attrition rates.

The postprocessor at Appendix E, developed for the homogeneous model, should provide a good foundation for the more complicated postprocessor needed for DAMSTAC II.

VII. CONCLUDING REMARKS

The concept of using differential equations to represent combat began with Lanchester's classical equations and has developed into a very dynamic field of study. The theoretical background of the fitted parameter method has been well documented by Clark and Taylor. However, the practical application of the theory lacks this same degree of discussion and documentation. This paper has focussed on some of the techniques of applying Clark's theory to an actual high resolution, Monte Carlo simulation. The problems are real and, as has been seen, the solutions presented are not doctrine. It is hoped that further investigation, elaboration and documentation of this modelling technique will occur in the near future.

The choice of which functional form to use in a particular analytical combat model is entirely up to the discretion of the model builder. In choosing an attrition rate functional form the model builder should consider the scenario portrayed in the model and the assumptions that are compatible with it.

Finally, it is recommended that the entire program for the model be converted to the UCSD Pascal language on the

Apple II. The basic structure of Pascal is more conducive to combat modelling simulation than the structure of the BASIC language. The concept of entities and attributes found in SIMSCRIPT II.5 can be closely emulated in Pascal while this is not as easily accomplished or clearly defined in BASIC.

The modular structure of procedures and functions in Pascal is also an advantage in the enhancement process of model building. Modellers can use one procedure to represent a particular action or phenomenon occurring on the battlefield. Any action which is unable to be modelled can be left in the program as a dummy procedure for later work.

Additionally, the Pascal language allows more descriptive variables to be used than the two-character variables used in Applesoft. This would make the program itself more transparent and easier to modify and update.

The DAMSTAC model is clearly in its infancy stage. Through continued enhancement, this model can contribute to the ongoing work in the field of combat analysis.

APPENDIX A

LIST OF VARIABLES

A = casualty rate for RED attritting acquired BLUE targets

AN = angle in radians measured counter-clockwise from due east made by the line between the current unit location and the next coordinating point.

B = casualty rate for BLUE attritting acquired RED targets

BE = the breakpoint for the BLUE force

BM = movement criteria for BLUE units as a fraction of unit strength

BF(I) = number of RED targets being fired at by BLUE unit I

BP(I) = most recently passed coordinating point for BLUE unit I

BR(I) = rate of movement for BLUE unit I in M/SEC

BT(I,J) = attrition(casualties) caused by RED unit J on BLUE unit I in one on one battle

BX(I,J) = x-coord of coordinating point J for BLUE unit I

BY(I,J) = y-coord of coordinating point J for BLUE unit I

B0(I) = current total BLUE force strength for time period I

B0(0) = initial BLUE force strength

B1 = number of coordinating points for BLUE (including

initial location)

B1(I) = initial strength of BLUE unit I

DB(I) = number of casualties or amount of attrition for BLUE unit I

DH = distance travelled by a unit in time increment DT, at speed RT

DI = distance between current unit location and next turn point

DR(J) = number of casualties or amount of attrition for RED unit J

DT = time increment

DX = x component of distance between current unit location and next coordinating point

DY = y component of distance between current unit location and next coordinating point

D1 = x component of distance moved in time increment DT, at speed RR(I) or BR(I)

D2 = y component of distance moved in time increment DT, at speed RR(I) or BR(I)

M = number of BLUE units

MT = max time of simulation

N = number of RED units

N$ = file name (used recursively)

P = probability that a BLUE target is not acquired by
a RED unit

PI = 3.141592654

PR = status of printer; 0 = printer is off
1 = printer is on

Q = probability that a RED target is not acquired by
a BLUE unit

RE = the breakpoint for the RED force

RF(J) = number of BLUE targets being fired at by RED unit J

RG = range between a BLUE unit and a RED unit

RM = movement criteria for RED units as a fraction of
unit strength

RP(I) = most recently passed coordinating point for RED
unit I

RR(I) = rate of movement for RED unit I in M/SEC

RT(J,I) = attrition (casualties) caused by BLUE unit I on
RED unit J in one on one battle

RX(I,J) = x-coord of coordinating point J for RED unit I

RY(I,J) = y-coord of coordinating point J for RED unit I

R0(I) = current total RED force strength for time period I

R0(0) = initial total RED force strength

R1 = number of coordinating points for RED (including
initial location)

R1(I) = initial strength of RED unit I

SB(I) = size of BLUE unit I

SR(I) = size of RED unit I

T = simulation time

XB(I) = current x-coord for BLUE unit I

XR(I) = current x-coord for RED unit I

YB(I) = current y-coord for BLUE unit I

YR(I) = current y-coord for RED unit I

APPENDIX B

NOTES ON THE COMPUTER MODEL

1. Data which have been hardwired into the model are:

A = 2.4

B = 0.03

BE = 0.3

BT = 0

DT = 30

MT = 1500

P = 0.99

PI = 3.141592654

Q = 0.93

RE = 0.3

2. Once contact is made (range ≤ 2700), the rate of advance decreases to 2 meters/second.

3. The option exists within the program to allow the movement of units based on a criteria of fractional force strength. When unit strength goes to 50% (movement criteria), the unit will move to an alternate location (for BLUE units) or retreat along its path of advance (for RED units).

APPENDIX C

UTILITY PROGRAMS

This appendix provides a brief description of the utility programs associated with the Aggregated Model of STAR on the Apple II Computer. These programs provide the capability to the user to transfer the necessary data from the STAR model to this model with the minimum effort.

The utility programs available are:

Force Maker

Results Reader

Both programs are interactive with the user, who answers questions provided by the computer. These programs use the commands for the Apple II Disk Operating System (DOS) which allow them to read/write Sequential Text Files and Random Access Files onto the disk. Familiarity with these types of files and the DOS would improve the understanding of these utility programs. [Ref. 13]

In the following program descriptions, the variable name used in the program is listed in parentheses after its description.

A. FORCE MAKER PROGRAM

This program will write data files containing information about the two oppcsing forces.

Force Maker produces a Sequential Text File containing:

Force size (number of units) (M)

The strength of each unit (SB(I))

The number of coordinating points for movement of each unit (the first coordinating point is always the initial location of the unit) (B1(I))

The grid coordinates of each coordinating point for each unit (BX(I,J), BY(I,J))

The user has the option to make a file for the BLUE force or the RED force. After all the data has been entered, a data check routine will print the information for all units, so that a check for errors can be made. After the user has checked the data and identified the name for the file, the program will record the file on the diskette.

A copy of the program is included as Listing B in Appendix F and a sample session using this program is attached as Figure 12.

```
]RUN FORCE MAKER
MENU:
1 MAKE FILE FOR BLUE FORCES
2 MAKE FILE FOR RED FORCES
0 QUIT

WHICH? 1
WHAT IS THE FORCE SIZE OF BLUE 3

FOR UNIT 1
HOW MANY COORDINATING POINTS (INCLUDING START LOCATION)? 1

ENTER GRID COORDS FOR COORDINATING POINT # 1
X-COORD IS 54163
Y-COORD IS 100655

FOR UNIT 2
HOW MANY COORDINATING POINTS (INCLUDING START LOCATION)? 1

ENTER GRID COORDS FOR COORDINATING POINT # 1
X-COORD IS 54788
Y-COORD IS 98120

FOR UNIT 3
HOW MANY COORDINATING POINTS (INCLUDING START LOCATION)? 1

ENTER GRID COORDS FOR COORDINATING POINT # 1
X-COORD IS 57448
Y-COORD IS 97993

            *** DATA CHECK ***


UNIT PT X-COORD Y-COORD
 1   1   54163   100655
 2   1   54788   98120
 3   1   57448   97993
ANY CHANGES? (Y/N) N
NAME OF FILE? THESIS
MENU:
1 MAKE FILE FOR BLUE FORCES
2 MAKE FILE FOR RED FORCES
0 QUIT

WHICH? 0
```

Figure 12: Sample Session of Force Maker Program

B. RESULTS READER PROGRAM

This program reads the data file of time intervals and corresponding BLUE and RED force levels produced by the model. Results Reader produces two sequential text files: the first file with suffix ".COEFS" contains:

Conditional Casualty rate, α (A)

Conditional Casualty rate, β (B)

Probability that a BLUE target is not acquired by a RED firer (P)

Probability that a RED target is not acquired by a BLUE firer (Q)

The second file with suffix ".RESULTS" contains:

The number of data points for BLUE force level versus time (N)

The number of data points for RED force level versus time (N)

Time of the battle at time interval I (T)

BLUE force level corresponding to time interval I (BO(I))

Time of the battle at time interval I (T)

RED force level corresponding to time interval I (RO(I))

The ".RESULTS" file contains some duplication of data for the following reasons. With the ".RESULTS" file written in this format, and free of additional data such as that in the

".COEFS" file, it can be used directly by the Apple Plot Program to produce a plot of the force levels versus time. A listing of the program is included as Listing C in Appendix F. A sample plot of the DAMSTAC output data produced by the Apple Plot Program is included as Figure 13. A sample of the output of DAMSTAC from Results Reader is included as Figure 14.

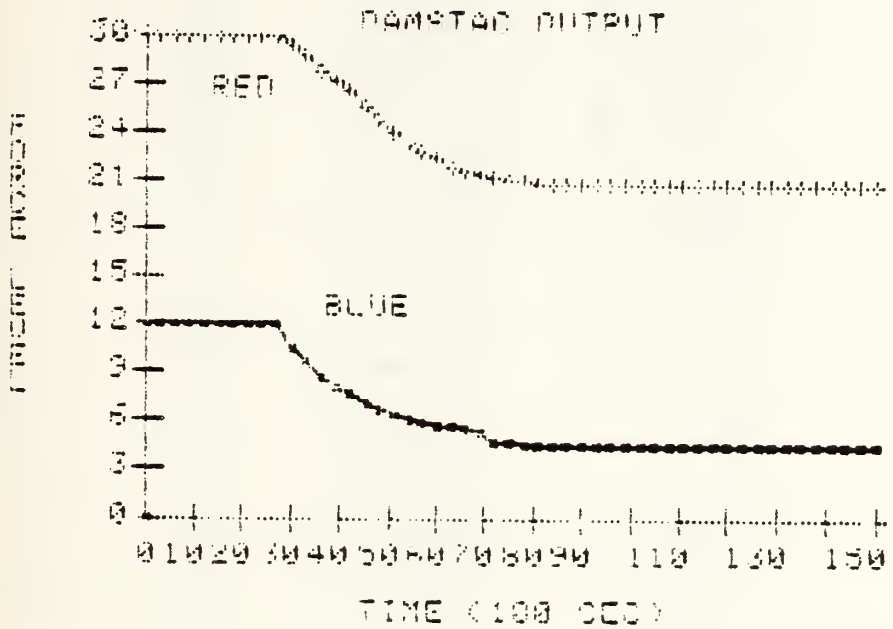


Figure 13: Sample Plot of Results Reader Program Output

```
A  =  1.3
B  =  .00
C  =  .00
D  =  .33

TIME     BLUE              RED
0        12                30
30       12                30
60       12                30
90       12                30
120      12                30
150      12                30
180      12                30
210      12                30
240      12                30
270      12                30
300      10.50026484       29.21586531
330      9.89579136        28.7382046
360      9.56389142        27.8546808
390      9.00097483        27.052176
420      7.86472383        25.577368
450      7.08704086        25.3177348
480      6.70180572        24.8362284
510      6.38847807        23.8943327
540      6.12758523        23.2837501
570      5.9087524         22.7232327
600      5.72306856        22.2585234
630      5.5642175         21.8509775
660      5.38920891        21.5240217
690      5.23620713        21.424205
720      4.99145474        21.1161326
750      4.89005959        20.8828377
780      4.48858501        20.8620825
810      4.4156453         20.7521797
840      4.4156453         20.7521797
870      4.4156453         20.7521797
900      4.4156453         20.7521797
930      4.4156453         20.7521797
960      4.4156453         20.7521797
990      4.4156453         20.7521797
1020     4.4156453         20.7521797
1050     4.4156453         20.7521797
1080     4.4156453         20.7521797
1110     4.4156453         20.7521797
1140     4.4156453         20.7521797
1170     4.4156453         20.7521797
1200     4.4156453         20.7521797
```

Figure 14: Sample Output from Results Reader Program

APPENDIX D

THE TERRAIN MODEL

In addition to the basic computer model, work was done to input the STAR terrain data base into the Apple II computer. This effort was done during the initial phase of the thesis work based on an erroneous assumption that the authors made concerning the functional form of the attrition rates. This assumption was that the terrain of the battlefield and line of sight computations must be explicitly played in the model. In the given functional forms for the attrition rates, line of sight and terrain are accounted for by the probability values of p and q.

Before this assumption was found to be false, four utility programs, and a model that computed the elevations of the units were developed and are included in this document for possible future use. If further work is done with a model on the Apple II computer, the ability to input the STAR terrain into the Apple II may become an important part of the model. Examples of models that might use the STAR terrain in the Apple II are small aggregated models with functional forms that explicitly play line of sight, or

small, high-resolution models that are suitable on the Apple II computer.

A. THE TERRAIN UTILITY PROGRAMS

The utility programs that allow the STAR terrain to be input in the Apple II computer are listed below.

Map Maker

Hill Maker

Woods Maker

Forest Maker

These four programs have two portions each: a file maker and a file reader. In general, the file maker portion allows the user to input a new file or update/change an old file. The file reader allows the user to print out on the screen or to a printer the selected file.

For the Apple II combat model, a STAR terrain file (eg. FULTERR file) is divided into its four main parts:

the list of hill reference numbers in each grid square;

the hill parameters;

the list of forest reference numbers in each grid square;

the forest ellipse fitting parameters.

Each part is created, respectively, by one of the four utility programs: Map Maker, Hill Maker, Woods Maker, and Forest

Maker. When these parts are being created on the Apple Disk, the same general file name must be used to identify the data's terrain (eg. FULDA). Each of the utility programs will add its own suffix to the general file name. For example, for the Fulda terrain, the four utility programs will produce files named:

FULDA.MAP

FULDA.HILLS

FULDA.WOODS

FULDA.FORESTS

This suffix identifies the file as a particular portion of the terrain data, and will allow the Apple computer to recognize it from the other files with the same terrain name when reading the file for the combat model.

1. Map Maker Program

This program will read and write data files containing a catalog by grid square of the hills which influence the terrain in that grid square. This program produces an equivalent file to that in STAR called LIST.H. Map Maker produces a Sequential Text File containing:

The size of the battlefield (normally 10 by 10) (L,W)

The lower left grid square in 1000 meter units (LX,LY)

The base elevation of the battlefield (HL(0,0,0))

For each grid square I,J:

(1) The number of hills influencing the terrain (HL(I,J,0))

(2) The reference number of these hills (HL(I,J,K))

The user has the option to make a new file or access an old file for updating or changes. For new files, the program will start with the lower left grid square and proceed from west to east then south to north. The data can be saved after the information for a grid square has been entered. For old files, the program will access it, print out the file on the screen and ask for the starting grid square. The grid square index is used to identify in which grid square to start. The grid square index is an integer from 1 to L in the east-west direction, and 1 to W in the south-north direction of an L by W sized battlefield. The lower left grid square is 1,1; the lower right is L,1; the upper left is 1,W; and the upper right is L,W.

A copy of the program is attached as Listing D in Appendix F and a sample file is attached as Figure 15.

```
]PR#0NAME OF MAP: FULDA
LOWER LEFT GRID SQUARE: 50 93
SIZE OF MAP: 10 BY 10
BASE ELEVATION: 300

1                    1                    3
22 25 26
1                    2                    3
26 25 125
1                    3                    5
79 126 80 111 26
1                    4                    6
127 78 36 128 111 130
1                    5                    5
37 128 130 36 38
1                    6                    3
128 74 131
1                    7                    7
74 122 115 116 114 132 44
1                    8                    5
116 115 120 121 122
1                    9                    6
115 144 10 116 118 53
1                    10                   4
54 9 144 115

2                    1                    3
22 27 26
2                    2                    5
25 27 26 126 112 111
2                    3                    4
111 126 112 25
2                    4                    7
79 111 36 37 134 127 128
2                    5                    7
36 37 75 128 134 74 72
2                    6                    7
74 114 72 73 128 75 76
2                    7                    5
44 114 143 115 74
2                    8                    7
122 142 91 115 123 121 44
2                    9                    7
119 127 122 119 144 115 142
2                    10                   4
108 118 54 144
```

Figure 15: Sample File for Map Maker Program

2. Hill Maker Program

This program reads/writes data files containing the fitting parameters for each hill on the battlefield. Hill Maker produces a Random Access Text File containing:

The number of hills on the battlefield (N)

For each hill I:

(1) X coord of the center location of the hill (H(I,1))

(2) Y coord of the center location of the hill (H(I,1))

(3) The elevation of the hill top measured from zero=sea level (H(I,3))

(4) The orientation angle of the ellipse measured in degrees counter-clockwise from east to the major axis. (H(I,4))

(5) The eccentricity, defined as the ratio of major axis length to minor axis length (H(I,5))

(6) The spread, defined as the distance in meters measured along the major axis from hill center to the contour line which is 50 meters down from the peak (H(I,6))

(7) The maximum height of the normal curve describing this hill mass (H(I,7))

(8) The vertical distance measured from the peak beyond which this hill is not considered in computations (H(I,8))

A copy of the program is included as Listing E in Appendix F, and a sample file is attached as Figure 16.

3. Woods Maker Program

This program will read and write data files containing a catalog by grid square of the woods which influence the terrain in that grid square. This program produces an equivalent file to that of Map Maker. Woods Maker produces a Sequential Text File containing:

The size of the battlefield (normally 10 by 10) (L,W)

The lower left grid square in 1000 meter units (LX,LY)

For each grid square I,J:

(1) The number of forests influencing the terrain (FL(I,J,0))

(2) The reference number of these forests (FL(I,J,K))

Woods Maker contains the same options as those found in Map Maker. A copy of the program is attached as Listing F in Appendix F, and a sample file is attached as Figure 17.

4. Forest Maker Program

This program reads/writes data files containing the fitting parameters for each forest on the battlefield. Forest Maker produces a Random Access Text File containing:

The number of forests on the battlefield (N)

For each forest I:

```
+RUPR+0
FULDA HILL DATA

NUMBER OF HILLS: 159

1       533     950     425     143     2       1250    350     85.75
2       540     959     430     114     3       800     250     76.85
3       540     950     430     45      2       500     250     61.11
4       542     933     430     74      2       800     180     106.16
5       544     954     435     135     2       500     250     87.58
6       549     948     450     0       1       500     300     106.18
7       559     946     460     0       1       300     350     45.53
8       563     939     450     263     3       900     500     120.54
9       505     1025    471     48      2       300     400     122.71
10      505     1014    390     48      1       300     140     30.55

11      573     960     460     40      3       900     210     95.71
12      544     1022    350     25      2       400     300     59.02
13      532     1023    325     62      3       1500    300     34.76
14      570     1019    415     130     2       1000    170     78.11
15      583     1025    390     14      2       800     250     88.93
16      595     1013    475     117     2       900     225     184.88
17      560     1009    390     4       2       1300    300     99.34
18      594     1002    460     137     1       500     225     61.75
19      573     1003    385     126     1       500     150     42.87
20      584     997     435     25      1       350     185     56.48

21      532     984     490     1       1       500     240     132.3
22      520     930     355     165     4       2000    140     64.94
23      581     1000    394     140     2       1750    100     39.91
24      584     1005    394     145     4       1000    100     42.71
25      524     949     380     200     4       2000    140     89.86
26      508     943     340     165     3       2000    100     49.92
27      530     945     390     27      2       1300    140     93.12
28      545     960     390     350     2       1000    150     63.15
29      543     984     390     89      4       1000    150     60.68
30      545     990     364     0       1       1000    100     57.97

31      551     981     400     210     2       1000    150     71.41
32      590     983     590     125     3       500     400     169.51
33      590     957     560     85      3       400     310     130.12
34      596     959     505     85      3       1500    300     196.2
35      596     959     505     120     2       1500    300     185.93
36      511     973     370     0       1       500     300     73.31
37      511     973     370     340     2       750     120     79.55
38      509     975     340     120     3       600     90      25.72
39      572     973     470     290     2       600     230     54.02
40      577     975     470     345     2       300     230     70.5
```

Figure 16: Sample File for Hill Maker Program

(1) X coord of the center location of the forest (F(I,1))

(2) Y coord of the center location of the forest (F(I,1))

(3) The height of the trees in the forest (F(I,3))

(4) The orientation angle of the ellipse describing the forest measured in degrees counter-clockwise from east to the major axis (F(I,4))

(5) The length of the semi-major axis of the ellipse (F(I,5))

(6) The length of the semi-minor axis of the ellipse (F(I,6))

Forest Maker contains the same options as those found in Hill Maker. A copy of the program is included as Listing G in Appendix F, and a sample file is attached as Figure 18.

B. THE TERRAIN MODEL PROGRAM

A copy of the model which explicitly plays terrain is included in Appendix F as Listing H. A line drawn along the edge of the program delineates the terrain model portion which was added to the basic model. The additional variables required to run the terrain model are also included at

```
NAME OF MAP: FULDA
LOWER LEFT GRID SQUARE: 50 33
SIZE OF MAP: 10 BY 10

1                    1                    1
39
1                    2                    1
59
1                    3                    1
23
1                    4                    2
23 53
1                    5                    4
65 62 50 53
1                    6                    3
63 97 50
1                    7                    2
96 97
1                    8                    2
94 95
1                    9                    1
94
1                    10                   3
17 95 94

2                    1                    1
58
2                    2                    2
8 50
2                    3                    2
8 23
2                    4                    2
23 53
2                    5                    5
5 26 65 53 50
2                    6                    3
65 97 50
2                    7                    5
94 96 98 99 97
2                    8                    4
95 94 98 96
2                    9                    1
94
2                    10                   2
17 94

3                    1                    2
```

Figure 17: Sample File from Woods Maker Program

the end of this appendix. Professor James K. Hartman [Ref. 14] provides the details of the methodology for line of sight and elevation using the STAR terrain.

equivalent STAR variables are in ( )

A1 = computing parameter for a hill (A)

B1 = computing parameter for a hill (B)

CR = critical value parameter for a hill (CRIT.H)

C1 = conversion of orientation angle to radians

C2 = (spread of a hill) squared

FI = value of the elevation of a point due to a hill (FI)

H(I,J) = the J-th parameter for hill I; the parameters are:

1 = x-coord of the center of the hill (XC.H)

2 = y-coord of the center of the hill (YC.H)

3 = peak elevation of the hill (PEAK.H)

4 = orientation angle of the hill (ANG.H)

5 = eccentricity of the hill (ECC.H)

6 = spread of the hill (SPRD.H)

7 = max height of the hill (HT.H)

8 = cut height of the hill (CUT.H)

H(I,0) = status of hill I;

0 = hill I has not been checked for LOS

1 = hill I has been checked for LOS

```
←PR#0
FULDA FOREST DATA

NUMBER OF FORESTS 109

1       558.75    991.25    3.75     90      250       187.5
2       562.27    982.31    16.25    25      250       125
3       570.89    957.49    8.75     85      375       250
4       544.68    946.2     28.75    84      875       562.5
5       515       973.63    28.75    0       312.5     187.5
6       531.25    946.88    28.75    0       750       375
7       536.09    958       28.75    100     812.5     312.5
8       525.56    954.94    28.75    45      875       250
9       537.25    971.25    28.75    0       100       50
10      551.17    949.88    28.75    140     500       50

11      560.85    944.85    23.75    70      250       156.25
12      593.71    961.61    23.75    140     250       156.25
13      588.74    994.74    23.75    -2      750       250
14      524.38    1002.75   23.75    90      150       50
15      583.4     953.11    23.75    85      312.5     250
16      588.02    951.98    23.75    17      250       100
17      509.84    1031.1    21.25    17      375       187.5
18      579.97    1020.55   21.25    5       750       375
19      564.45    1029.09   21.25    -25     1250      500
20      582.17    952.4     21.25    -12     350       50

21      530.62    952.14    21.25    4       375       187.5
22      583.51    1013.17   18.75    -34     625       437.5
23      514.99    958.02    18.75    3       1000      500
24      523.97    960.38    18.75    -52     375       187.5
25      529.16    966.12    18.75    -3      250       250
26      516.78    976.03    18.75    -23     375       187.5
27      554.85    981.64    18.75    -20     250       250
28      543.75    992.43    18.75    -1      375       187.5
29      570       1012.25   18.75    90      100       100
30      572.5     1028.75   18.75    0       250       100

31      596.25    997.88    18.75    90      100       100
32      588.72    1014.86   18.75    -28     562       281.25
33      596.48    1008.2    18.75    117     500       312.5
34      520.25    995       18.75    0       150       50
35      521.25    1004      18.75    90      150       150
36      559       936.78    18.75    24      437       312.5
37      535.25    962.5     18.75    90      375       250
38      548.37    972.35    18.75    -2      437.5     250
39      501.74    934.89    18.75    130     312.5     187.5
40      596.25    946.88    18.75    90      187.5     187.5
```

Figure 18: Sample File from Forest Maker Program

HL(I,J,K) = the number of the K-th hill in grid square I,J which influences the terrain in that grid square I = 1 to L; J = 1 to W; K = 1 to HL(I,J,0)

HL(I,J,0) = the total number of hills in grid square I,J which influence the terrain in that grid square

HL(0,0,0) = the base elevation of the terrain map

H2 = temporary variable identifying the hill being checked for line of sight

L = number of grid squares on the terrain map in the x direction (NGRIDX)

LX = x-coord of lower left grid square of terrain map in 1000 meter units (X.LO.BDRY)

LY = y-coord of lower left grid square of terrain map in 1000 meter units (Y.LO.BDRY)

M$ = name of files containing terrain data

NH = total number of hills on the terrain map (NHILLS)

P1 = computing parameter for a hill (PXX.H)

P2 = computing parameter for a hill (PXY.H)

P3 = computing parameter for a hill (PYY.H)

QI = quadratic function of the ellipse represnting the hill (QI)

W = number of grid squares on the terrain map in the Y direction (NGRIDY)

X = current x-coord of unit whose elevation is being computed (in 100 meter units) (X)

XB(I) = current x-coord for BLUE unit I

XR(I) = current x-coord for RED unit I

XS = distance in x direction between unit location and center of hill mass (XS)

X1 = relative number of grid squares in x direction of a BLUE unit location with respect to the lower left grid square

X2 = relative number of grid squares in x direction of a RED unit location with respect to the lower left grid square

X3 = number of grid squares in x direction between a BLUE unit and a RED unit

X4,Y4 = current grid square being checked for line of sight in LOS subroutine

Y = current y coord of unit whose elevation is being computed (in 100 meter units) (Y)

YB(I) = current y-coord for BLUE unit I

YR(I) = current y-coord for RED unit I

YS = distance in y direction between unit location and center of hill mass (YS)

Y1 = relative number of grid squares in y direction of a BLUE unit location with respect to the lower left grid square

Y2 = relative number of grid squares in y direction of a RED unit location with respect to the lower left grid square

Y3 = number of grid squares in y direction between a BLUE unit and a RED unit

Z = temporary variable for the elevation at a point on a hill (Z)

ZB(I) = elevation of BLUE unit I

ZR(J) = elevation of RED unit J

APPENDIX E

SYNOPSIS OF THE POSTPROCESSOR

The following appendix is a summary of the work done by other students as a part of a joint project for OA 4655. The authors gratefully acknowledge the work done by Captains Ambrose R. Hock and Steven L. Maddox in developing this postprocessor to reduce the amount of work required in extracting the necessary data from STAR. Their work is included here in this appendix since it is an integral part of the parameter estimation process.

The postprocessor developed for this thesis is a Simscript language program that provides to the user a summary of the data generated from the Simulation of Tactical Alternative Responses(STAR) Model. Additionally, the postprocessor performs some routines that assist the user in the data analysis of the model. The STAR Model is a Simscript language program that simulates combat between two combined arms teams in a combat environment that includes field artillery fire and electronic warfare.

Since the model assumed in the project only required data generated by a tank vs tank battle, it was necessary to

change certain aspects of the STAR model. Thus it was necessary to "turn off" the field artillery and Electronic Warfare modules within the STAR model. Also the non-tank units needed to be deleted. Referring to figure 19, the Electonic Warfare code(EWCODE) was deleted through the use of a comment card, and a Electonic Warfare off(EWOFF) subroutine was inserted.

```
//STARTHES JOB (3102,0234),'COMB ARMS BATTLE',CLASS=C
//SIM.SYSPRINT DD  DUMMY
//SIM.SYSIN  DD  DISP=SHR,DSN=MSS.S3102.THESIS(WRKPREAM)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(SNAPDOTR)
//*          DD  DISP=SHR,DSN=MSS.S3102.THESIS(EWCODE)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(MOVCOORD)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(COMMWORK)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(TEMPCOMM)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(TEMPGRND)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(EWOFF)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(FAEXPER)
//           DD  DISP=SHR,DSN=MSS.S3102.THESIS(ARTYMESG)
```

Figure 19: Electronic Warfare Deletion

Thus, whenever the main program called for the usage of EW, the EWOFF subroutine automatically returned the program back to the point where the EW routine was called and the simulation continued with no Electronic Warfare generated. The field artillery within the STAR model was cancelled by

having the simulation create FA fire at time equal to 5000 time units(see Figure 20).

Schedule a FABEGIN in 5000.0001 time units

Figure 20: Field Artillery Deletion

Since the simulation was scheduled to run only 2500 time units, no field artillery was generated. Finally, the internal logic of the STAR model was used to "destroy" the non-tank units within the simulation by changing their basic load to zero(see Figure 21). This action created situations, where the tank elements did not consider the non-tank elements as lethal or dangerous targets and thus were ignored.

```
1 1    1  1 0 38    1  1 0 17    1 1 0 100    0 0 0 0
2 3    0  0 0 0     0  0 0 0     0 0 0 0      0 0 0 0
2 4    1  0 1 10    0  0 0 0     0 0 0 0      0 0 0 0
3 6    0  0 0 0     1  0 1 2     0 0 0 0      0 0 0 0
4 1   50 20 0 50   60 30 0 50    0 0 0 0      0 0 0 0
1 7    1  1 0 22    1  1 0 18    1 1 0 100    0 0 0 0
2 8    1 0  1 4     0  0 0 0     1 1 0 11     0 0 0 0

1 1    1  1 0 38    1  1 0 17    1 1 0 100    0 0 0 0
2 3    0  0 0 0     0  0 0 0     0 0 0 0      0 0 0 0
2 4    0  0 0 0     0  0 0 0     0 0 0 0      0 0 0 0
3 6    0  0 0 0     0  0 0 0     0 0 0 0      0 0 0 0
4 1   50 20 0 50   60 30 0 50    0 0 0 0      0 0 0 0
1 7    1  1 0 22    1  1 0 18    1 1 0 100    0 0 0 0
2 8    0  0 0 0     0  0 0 0     0 0 0 0      0 0 0 0
```

Figure 21: Alteration of Basic Load

Output from the STAR model can be divided into three categories:

Update list of attacker/defender status.

Shot data

Significant event listing (i.e. movement, EW and FA events)

Only the shot data from the STAR model is needed to provide the necessay data for computations. Figure 22 is a sample of this output. In order to utilize this data, the internal output control of the model was used to place the shot data in a mass storage location within the computer to be later manipulated by the postprocessor.

The flow chart for the postprocessor is contained in Figure 23; a copy of the program itself can be found in Appendix F as Listing I. The program flow is very basic and needs little explanation. From the mass storage location of the computer, the postprocessor reads pertinent information into the SHOTLIST. Once all of the data has been read, the program processes the data into a casualty list(CASLIST) keying from a status of "dead" from the Shotlist. Figure 24 is a sample of the casualty list output. Once CASLIST is created the postprocessor performs histogram computations and graphs. The postprocessor can print two types of histograms. Figure 25 contains the Simcsript Library program for the histogram. Additionally, the HISTG routine in

```
FILE: TEMPSUM  FILE     3   NAVAL POSTGRADUATE SCHOOL

  1  257 2704   38 1-7 1   30 1-1   .26 57391 100581 377   5.0  5 57730  97900 439
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
  2  266 2610   30 1-1 1   30 1-7 1.00 57730  97900 439   0.   3 57456 100495 380
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
  3  276 2561   28 1-1 1   30 1-7   .30 57320  97900 427   0.   3 57435 100458 380
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
  4  289 2533   27 1-1 1   31 1-7 1.00 57140  97920 428   0.   3 57476 100430 382
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
  5  303 2458   33 1-7 1   28 1-1   .07 57543 100347 384   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
  6  307 2507   29 1-1 1   31 1-7 1.00 57600  97850 433   0.   3 57431 100350 380
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
  7  316 2275   44 1-7 1   28 1-1   .07 57957 100084 393   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
  8  318 2390   63 1-7 1   28 1-1   .07 57506 100282 383   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  3  0
  9  321 2391   62 1-7 1   28 1-1   .07 57452 100287 381   5.0  5 57320  97900 427
 0.    2 1.80   MEDM 0.    0.  0.    1 1 1 0   0  4  1
 10  322 2252   45 1-7 1   28 1-1   .07 57996 100048 394   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    1 1 1 0   0  5  1
 11  324 2236   46 1-7 1   28 1-1   .07 58036 100017 394   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    1 1 1 0   0  6  1
 12  328 2305   64 1-7 1   30 1-1   .36 57513 100194 383   5.0  5 57730  97900 439
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
 13  332 2335   30 1-1 1   63 1-7 1.00 57730  97900 439   0.   3 57471 100220 381
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 14  334 2185   44 1-7 1   28 1-1   .07 57920 100001 392   5.0  5 57320  97900 427
 0.    2 2.02   MDAM 0.    0.  0.    1 1 1 0   0  7  2
 15  335 2314   63 1-7 1   28 1-1   .07 57465 100209 381   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    1 1 1 0   0  8  2
 16  340 2154   43 1-7 1   28 1-1   .07 57858  99985 391   5.0  5 57320  97900 427
 0.    2 3.00   MDAM 0.    0.  0.    1 1 1 0   0  9  3
 17  341 2150   45 1-7 1   28 1-1   .07 57958  99964 393   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    1 1 1 0   0 10  3
 18  344 2136   46 1-7 1   28 1-1   .07 57995  99926 393   5.0  5 57320  97900 427
 0.    2  0.    MISS 0.    0.  0.    1 1 1 0   0 11  3
 19  355 2118   27 1-1 1   44 1-7   .68 57140  97920 428   0.   3 57877  99905 390
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 20  356 2076   44 1-7 1   28 1-1   .07 57875  99900 389   5.0  5 57320  97900 427
 0.    2 2.02   MDAM 0.    0.  0.    1 1 1 0   0 12  4
       381      EXPO       28 1-1                                    57320  97900 427
 0.    2        DEAD 0.    0.  0.    1 1 1 0   0 12  4
 21  436 2756    7 1-1 1   56 1-7 1.00 54150 100690 340   0.   3 56777  99858 355
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 22  438 2646   20 1-1 1   58 1-7   .98 54940  98000 378   0.   3 56869  99812 361
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 23  445 2625   55 1-7 1   29 1-1   .36 56724  99835 353   5.0  5 54190 100520 341
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 24  454 2629   56 1-7 1   29 1-1   .36 56716  99790 355   5.0  5 54190 100520 341
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
 25  454 2629    9 1-1 1   56 1-7   .37 54190 100520 341   0.   3 56715  99790 355
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
 26  455 2630    8 1-1 1   55 1-7   .42 54150 100620 341   0.   3 56700  99796 354
 5.0   5  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 27  455 1859   58 1-7 1   27 1-1   .36 56811  99748 362   5.0  5 57140  97920 428
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 28  467 1822   27 1-1 1   58 1-7   .97 57140  97920 428   0.   3 56770  99702 361
 5.0   5 .94    MEDM 0.    0.  0.    1 1 1 0   0  2  1
 29  469 2493   10 1-1 1   58 1-7   .95 54360  98100 377   0.   3 56770  99702 361
 0.    5 .13    DEAD 0.    0.  0.    1 1 1 1   1  3  2
 30  485 2647   63 1-7 1   20 1-1   .07 56902  99777 364   5.0  5 54940  98000 378
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  1  0
 31  486 2632   64 1-7 1   20 1-1   .07 56920  99734 367   5.0  5 54940  98000 378
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  2  0
 32  494 1433   42 1-7 1   30 1-1   .07 57559  99322 401   5.0  5 57730  97900 439
 0.    2  0.    MISS 0.    0.  0.    0 0 0 0   0  3  0
 33  496 2056   57 1-7 1   29 1-1   .07 56004  99646 351   5.0  5 57600  97850 438
 0.    2 1.24   MEDM 0.    0.  0.    1 1 1 0   0  1  1
 34  502 2557   54 1-7 1   30 1-1   .36 56446  99705 365   5.0  5 54940  98000 378
 0.    3  0.    MISS 0.    0.  0.    0 0 0 0   0  3  0
 35  504 2592   18 1-1 1   64 1-7 1.00 54740  98180 376   0.   3 56839  99702 365
```

Figure 22: STAR Output Shotlist

FORTRAN can be accessed to produce a histogram of the data. After the histogram data is generated, the postprocessor calculates the attrition rate coefficients using the formulas discussed in Section III-E of this thesis.

Brief description of the postprocessor (see this appendix for variable description):

Preamble:

lines 1 - 9	definition of permanent entities.
lines 10 - 39	definition of temporary entities.
lines 40 - 54	definition of data to be collected for histograms.

Main:

line 2	variable definition.
line 3 - 4	system parameters read.
lines 13 - 31	creation of shotlist, reading data from file.
lines 43 - 89	creation of casualty list, file "dead".
lines 90 - 125	processing of histogram data
lines 125 - 186	processing of attrition rate coefficients.

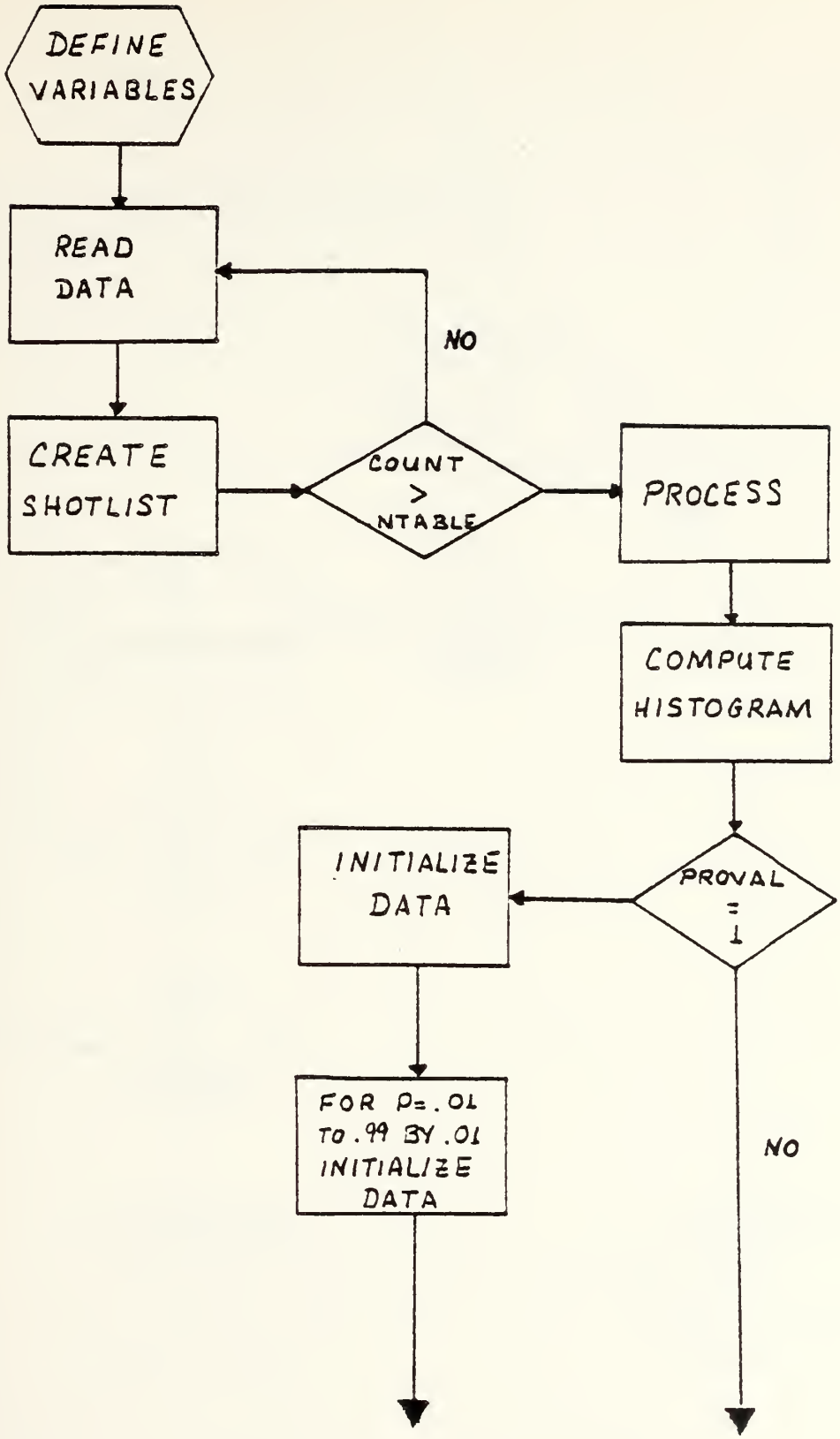
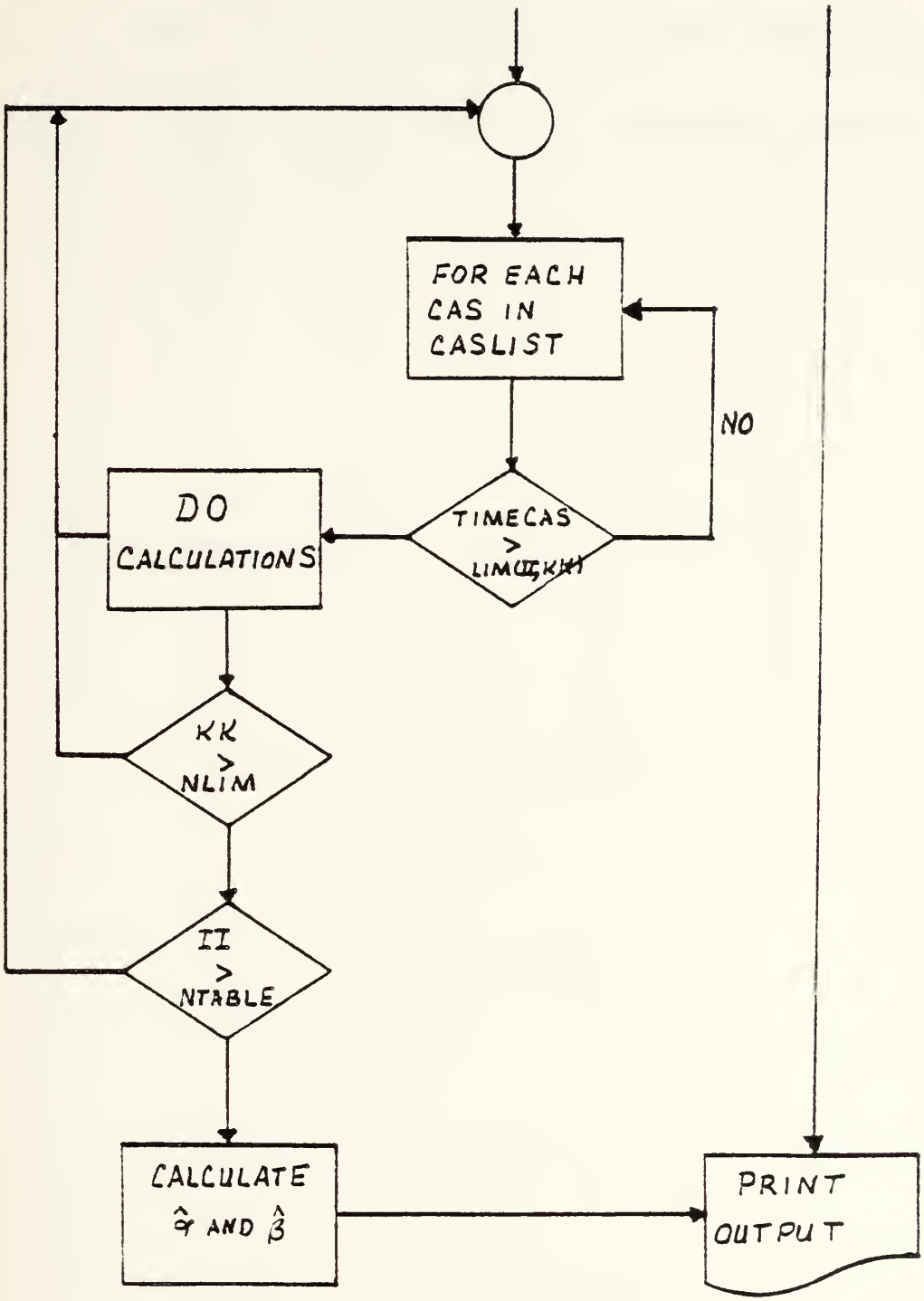


Figure 23: Flow Chart of the Postprocessor


FOR EACH CAS IN CASLIST
NO
TIMECAS > LIMC(II,KK)
DO CALCULATIONS
KK > NLIM
II > NTABLE
CALCULATE $\hat{\alpha}$ AND $\hat{\beta}$
PRINT OUTPUT

SHOT LIST DATA FOR CASUALTIES FOR THIS STAR SIMULATION

NUM	TIME	TARGET	SYS-WPN TYPE	TIME BETWEEN CASUALTIES
1	322	59	1-7	0
2	334	46	1-7	12
3	468	58	1-7	134
4	574	28	1-1	106
5	581	42	1-7	7
6	591	27	1-1	10
107	1210	72	1-7	6
108	1238	69	1-7	28
109	1265	73	1-7	27
110	1303	75	1-7	38
111	1323	19	1-1	20
112	1421	76	1-7	98

Figure 24: Casualty List

NUMBER OF BLUE KILLED = 28

BLMU = 47.25000 BLSIG = 299.259

NUMBER OF RED KILLED = 84

RDMU = 13.24096 RDSIG = 219.106

HISTOGRAM OF BLUE AND RED CASULTIES
(DATA BASED ON TIME BETWEEN BLUE CASULTIES
AND TIME BETWEEN RED CASULTIES)

INTERVAL=10 UNITS	#BLUE CAS	#RED CAS
1	3	11
2	4	8
3	4	20
4	0	4
5	0	8
6	0	4
7	0	0
8	0	8
9	0	0
10	0	0
11	0	4
12	4	8
13	0	4
14	4	4
15	0	0
16	0	0
17	0	0
18	0	0
19	4	0
20	0	0
21	5	0

Figure 25: Simscript Histogram

List of Variables For Postprocessor

AHAT = estimate of the attrition coefficient for RED attriting BLUE

BHAT = estimate of the attrition coefficient for BLUE attriting RED

BLHST = Simscript routine for the histogram of BLUE time between casualties

BLMU = Simscript routine for calculating the mean time

between casualties

BLSHTHST = Simscript routine for the histogram of BLUE shot times

BLSHTIG = Simscript routine for calculation of the standard deviation of BLUE shot times

BLSHTMU = Simscript routine for calculation of the mean of BLUE shot times

BLSIG = Simscript routine for calculation of the standard deviation of BLUE time between casualties

BLUE.SHOTS = time of the previous BLUE shot

BLUE.TIME = time of the previous BLUE casualty

CAS = casualty

CASHST = Simscript routine for the histogram of total time between casualties

CASIG = Simscript routine for calculation of the standard deviation of the total time between casualties

CASLIST = array containing list of casualties

CASMU = Simscript routine for calcualtion of the mean for the total time between casualties

CBLUE = number of surviving BLUE force elements

CK = integer used in the calculation of the attrition coefficient

1 = BLUE casualty

0 = RED casualty

CRED = number of surviving RED force elements

CURTIME = time of previous casualty

FNAME = firer's name

FSWTYPE = firer's system-weapon type

J.BL = number of BLUE casualties(used in BLHST routine)

J.RD = number of RED casualties(used in RDHST routine)

JBLHST = number of BLUE shots(used in BLSHTHST routine)

JCAS = number of casualties(used in CASHST routine)

JRDSHT = number of RED shots(used in RDSHTHST routine)

JSHOT = number of shots(used in SHOTHST routine)

JTOT = number of casualties(used in TOTHST routine)

LIM = array of time periods that represent phases of
the battle

LIMA = lower time limit of phase

LIMB = upper limit of phase

LP = partial derivative of p

LQ = partial derivative of q

MAXBLUE = initial BLUE force size

MAXRED = initial RED force size

MINP = minimum of p

MINQ = minimum of q

MK1 = number of RED elements in time interval

NK1 = number of BLUE elements in the time interval

NLIM = number of time intervals to be evaluated

NSHOT = number of shots in the total battle

NTABLE = number of tables(battles) to be evaluated

NUM = counter fo number of casualties

NUMBL.CAS = number of BLUE casualties

NUMRD.CAS = number of RED casualties

NUMCAS = casualty entity number

P = probability of not acqiring a target

PHAT = estimated p

PROVAL = process variable

0 = stop

1 = process

QHAT = estimate of q

RANGE = range of firer to target

RDHST = Simscript routine for the histogram of RED casualties

RDMU = Simscript routine for calulation of the average time between RED casualties

RDSHTHST = Simscript routine for the histgram of total red shots

RDSHTMU = Simscript routine for calculation of the average

red shot time

RDSHTSIG = Simscript routine for the calculation of the standard deviation of the RED shot time

RDSIG = Simscript routine for calculation of the standard deviation for RED time between casualties

RED.SHOTS = time of previous RED shot

RED.TIME = time of previous RED casualty

RUNNO = run number

SHOT = shot

SHOTHST = Simscript routine for total shots fired

SHOTLIST = array of shot times

SHOTMU = Simscript routine for calculation of the average time of shots

SHOTSIG = Simscript routine for calculation of the standard deviation of total shots

STATUS = status of shot

STATUS.CAS = status of casualty

T.BTWN.CAS = time between casualty

T.CAS = casualty number

TBLUE = number of BLUE casualties within the interval

TGT.CAS = target casualty

TIME = time of shot

TIMECAS = time of casualty

TIMELIM = maxtime of battle when a casualty occurred

TMCAS = time of previous casualty

TMRDCAS = time of previous RED casualty

TOTAL.CASTIME = total casualty time

TOTAL.SHOT.TIME = total shot time

TOTAL.TIME = total casualty time for the creation of
T.BTWN.CAS

TOTHST = Simscript routine for the histrogram of
total casualty time

TOTMU = Simscript routine for calculation of the
average casualty time

TOTSIG = Simscript routine for calculation of the
standard deviation of the total casualty time

TRED = total RED casualties within the time interval

TSUB = time counter

TSWTYPE = target system-weapon type

TTBLUE = total BLUE casualties for XX battles

TTRED = total RED casualties for XX battles

TTCAS = total casualties for XX battles

TTT = number of shot

TYPE.CAS = system weapon type of the casualty

VAR = attrition coefficient variables

APPENDIX F

LISTING OF COMPUTER PROGRAMS

A. DAMSTAC MODEL PROGRAM

```
10  REM  UPDATED 24 FEB 82
20  REM  DETERMINISTIC AGGREGATED MODEL OF STAR ON THE
21  REM  APPLE COMPUTER (DAMSTAC)
30  DIM XB(5),YB(5),XR(5),YR(5),BX(5,10),BY(5,10),RX(5,10)
40  DIM BP(5),RP(5),SB(5),SR(5),RY(5,10)
50  DIM BF(5),RF(5),BT(5,5),RT(5,5),DB(5),DR(5)
60  DIM BR(5),RR(5)
70  DIM B1(5),R1(5)
80 T = 0:DT = 30:MT = 1500
90 D$ =  CHR$ (4)
100 P = 0.99:Q = 0.93
110 PI = 3.141592654:PR = 0
120 A = 2.4:B = 1.2
130 BE = 0.3:RE = 0.3
140 BM = .5:RM = .5
150 KK = 0
160 NK = MT / DT
161  REM  ZZ=FLAG FOR TYPE OUTPUT
162  REM  ZZ=1 GIVES UNIT
163  REM  LOCATIONS & SIZES
164  REM  ZZ=0 GIVES TIME & TOTAL
165  REM  RED & BLUE FORCE LEVELS
166 ZZ = 0
170  DIM B0(NK),R0(NK)
180  HOME
181  REM  READ DATA FROM FILE FOR
182  REM  BLUE FORCE.
183  REM  DATA INCLUDES:
184  REM  M=NUMBER OF UNITS
185  REM  B1=NUMBER OF COORD PTS
186  REM  BX,BY=GRID OF COORD PT
187  REM
190  INPUT "NAME OF FILE FOR BLUE DATA? ";N$
200  PRINT D$;"OPEN ";N$;".BLUE"
210  PRINT D$;"READ ";N$;".BLUE"
220  INPUT M
230  FOR I = 1 TO M
240  INPUT B1
250  FOR J = 1 TO B1
260  INPUT BX(I,J): INPUT BY(I,J)
270  NEXT : NEXT
280  PRINT D$;"CLOSE ";N$;".BLUE"
```

```
281  REM  SET INITIAL UNIT
282  REM  LOCATION EQUAL TO
283  REM  1ST COORD PT.
284  REM
290  FOR I = 1 TO M:XB(I) = BX(I,1):YB(I) = BY(I,1):BP(I) = 1
300  PRINT B1: FOR JJ = 1 TO B1: PRINT BX(I,JJ),BY(I,JJ): NEXT JJ
310  PRINT "SIZE OF BLUE UNIT ";I;: INPUT SB(I)
320 B1(I) = SB(I)
321  REM
322  REM  SUM INITIAL FORCE SIZE
323  REM
330 B0(0) = B0(0) + SB(I)
331  REM
332  REM  RATE OF BLUE MOVEMENT
333  REM  EQUALS 0
334  REM
340 BR(I) = 0
350  NEXT
360  PRINT
361  REM
362  REM  READ DATA FROM FILE FOR
363  REM  RED FORCE.
364  REM  SAME FORMAT AS BLUE
365  REM
370  INPUT "NAME OF FILE FOR RED DATA? ";N$
380  PRINT D$;"OPEN ";N$;".RED"
390  PRINT D$;"READ ";N$;".RED"
400  INPUT N
410  FOR I = 1 TO N
420  INPUT R1
430  FOR J = 1 TO R1
440  INPUT RX(I,J): INPUT RY(I,J)
450  NEXT : NEXT
460  PRINT D$;"CLOSE ";N$;".RED"
470  FOR I = 1 TO N:XR(I) = RX(I,1):YR(I) = RY(I,1):RP(I) = 1
480  PRINT "SIZE OF RED UNIT ";I;: INPUT SR(I)
490 R1(I) = SR(I)
500 R0(0) = R0(0) + SR(I)
510 RR(I) = 5
520  NEXT I
530  PRINT : INPUT "DO YOU WANT THE PRINTER ON? (Y/N) ";A$
535  IF A$ = "Y" THEN PR = 1
540  HOME : VTAB 3
550  IF PR = 1 THEN  PR# 1
560  PRINT
564  REM
565  REM  PRINT INITIAL DATA
566  REM
570  PRINT "TIME";: POKE 36,7: PRINT "BLUE";: POKE 36,20: PRINT "RED"
```

```
580  PR# 0
584  REM
585  REM  WHICH OUTPUT TYPE?
586  REM
590  IF ZZ = 1 THEN  GOSUB 1760: GOTO 610
600  GOSUB 1490
610  REM  MAIN PROGRAM
620 T = T + DT
624  REM
625  REM  CHECK IF MAX TIME
626  REM  IS EXCEEDED
627  REM
630  IF T <  = MT THEN 640
635  PRINT "END OF BATTLE DUE TO TIME": GOSUB 1540: END
640  GOSUB 790: REM  MOVEMENT SUBROUTINE
650  GOSUB 1170: REM  ATTRITION SUBROUTINE
654  REM
655  REM  INCREMENT THE TIME
656  REM  PERIOD INDEX
657  REM
660 KK = KK + 1
670 B0(KK) = 0:R0(KK) = 0
674  REM
675  REM  SUM BLUE FORCE LEVEL
676  REM  FOR THIS TIME PERIOD
677  REM
680  FOR I = 1 TO M
690 B0(KK) = B0(KK) + SB(I)
700  NEXT I
704  REM
705  REM  SUM RED FORCE LEVEL
706  REM  FOR THIS TIME PERIOD
707  REM
710  FOR J = 1 TO N
720 R0(KK) = R0(KK) + SR(J)
730  NEXT J
734  REM
735  REM  WHICH TYPE OF OUTPUT?
736  REM
740  IF ZZ = 1 THEN  GOSUB 1760: GOTO 750
750  GOSUB 1490
```

```
752  REM
753  REM   CHECK IF END OF BATTLE
754  REM - CRITERIA FOR FORCE
755  REM   LEVELS IS EXCEEDED
756  REM  IF CRITERIA IS EXCEEDED
757  REM  GOTO SAVE RESULTS
758  REM
760  IF B0(KK) > BE * B0(0) THEN 770
765  PRINT : PRINT "END OF BATTLE DUE TO ATTRITION OF BLUE"
766  GOSUB 1540: END
770  IF R0(KK) > RE * R0(0) THEN 780
775  PRINT : PRINT "END OF BATTLE DUE TO ATTRITION OF RED"
776  GOSUB 1540: END
780  GOTO 620
790  REM  MOVEMENT SUBROUTINE
794  REM
795  REM  FOR RED UNITS
796  REM
800  FOR I = 1 TO N
804  REM
805  REM  GET LAST COORD PT
806  REM
810 J = RP(I)
813  REM
814  REM  COMPUTE COMPONENTS OF
815  REM  DISTANCE BETWEEN NEXT
816  REM  COORD PT AND CURRENT
817  REM  LOCATION
818  REM
820 DY = RY(I,J + 1) - YR(I)
830 DX = RX(I,J + 1) - XR(I)
833  REM
834  REM  DO GEOMETRY TO GET
835  REM  CORRECT ANGLE (AN)
836  REM
840  IF DX = 0 AND DY > 0 THEN AN = PI / 2: GOTO 880
850  IF DX = 0 AND DY < 0 THEN AN = 3 * PI / 2: GOTO 880
860 AN =  ATN (DY / DX)
870  IF DX < 0 THEN AN = PI + AN
873  REM
874  REM  COMPUTE DH, DISTANCE
875  REM  TO BE TRAVELLED IN
876  REM  THIS TIME PERIOD
877  REM
880 D1 = DT * RR(I) *  COS (AN)
890 D2 = DT * RR(I) *  SIN (AN)
900 DH =  SQR (D1 ^ 2 + D2 ^ 2)
```

```
904  REM
905  REM  COMPUTE DI, DISTANCE
906  REM  TO NEXT COORD PT
907  REM  FROM CURRENT LOCATION
908  REM
910 DI =  SQR (DX ^ 2 + DY ^ 2)
911  REM
912  REM  IF DI < DH, MOVE UNIT
913  REM  ONLY THE AMOUNT TO
914  REM  NEXT COORD PT
915  REM  IF DI IS USED, UPDATE
916  REM  LAST COORD PT
917  REM
920  IF DI >  = DH THEN 950
930 D1 = DX:D2 = DY
940 RP(I) = J + 1
950 YR(I) =  INT (YR(I) + D2)
960 XR(I) =  INT (XR(I) + D1)
970  NEXT
974  REM
975  REM  DO THE SAME FOR BLUE
976  REM
980  FOR I = 1 TO M
985  IF BR(I) = 0 THEN 1150
990 J = BP(I)
1000 DY = BY(I,J + 1) - YB(I)
1010 DX = BX(I,J + 1) - XB(I)
1020  IF DX = 0 AND DY > 0 THEN AN = PI / 2: GOTO 1060
1030  IF DX = 0 AND DY < 0 THEN AN = 3 * PI / 2: GOTO 1060
1040 AN =  ATN (DY / DX)
1050 IF DX < 0 THEN AN = PI + AN
1060 D1 = DT * BR(I) *  COS (AN)
1070 D2 = DT * BR(I) *  SIN (AN)
1080 DH =  SQR (D1 ^ 2 + D2 ^ 2)
1090 DI =  SQR (DX ^ 2 + DY ^ 2)
1100  IF DI >  = DH THEN 1130
1110 D1 = DX:D2 = DY
1120 BP(I) = J + 1
1130 YB(I) =  INT (YB(I) + D2)
1140 XB(I) =  INT (XB(I) + D1)
1150  NEXT
1160  RETURN
```

```
1170  REM   ATTRITION SUBROUTINE
1172  REM
1173  REM  ZERO OUT ATTRITION
1174  REM  VALUES FOR ALL UNITS
1175  REM
1180  FOR I = 1 TO M:BF(I) = 0:OB(I) = 0: NEXT
1190  FOR J = 1 TO N:RF(J) = 0:OR(J) = 0: NEXT
1192  REM
1193  REM  CHECK RANGE BETWEEN
1194  REM  OPPONENTS. IF RANGE
1195  REM  <= 2700 ATTRITION WILL
1196  REM  OCCUR. REDUCE RATE OF
1197  REM  RED MOVEMENT TO 2 M/S
1198  REM
1200  FOR I = 1 TO M
1210  FOR J = 1 TO N
1220 BT(I,J) = 0:RT(J,I) = 0
1230 RG =  SQR ((XB(I) - XR(J)) ^ 2 + (YB(I) - YR(J)) ^ 2)
1240  IF RG > 2700 THEN 1310
1250 RR(J) = 2
1251  REM
1252  REM  INCREMENT THE NUMBER
1253  REM  OF TGTS FIRED AT BY
1254  REM  EACH UNIT. COMPUTE
1255  REM  ATTRITION CAUSED BY
1256  REM  RED ON BLUE & VICE
1257  REM  VERSA
1258  REM
1270 BF(I) = BF(I) + 1
1280 RF(J) = RF(J) + 1
1285  IF RR(J) =  - 8 THEN BT(I,J) = 0: GOTO 1295
1290 BT(I,J) = A / N * (1 - (P) ^ SB(I)) * SR(J)
1295  IF SR(I) = 8 THEN RT(J,I) = 0: GOTO 1310
1300 RT(J,I) = B / M * (1 - (Q) ^ SR(J)) * SB(I)
1310  NEXT : NEXT
1311  REM
1312  REM  SUM TOTAL ATTRITION
1313  REM  FOR EACH UNIT
1314  REM
1320  FOR I = 1 TO M
1330  FOR J = 1 TO N
1340  IF RF(J) = 0 OR BF(I) = 0 THEN 1370
1350 OB(I) = OB(I) + BT(I,J) / RF(J)
1360 OR(J) = OR(J) + RT(J,I) / BF(I)
1370  NEXT : NEXT
```

```
1371  REM
1372  REM  COMPUTE NEW UNIT
1373  REM  STRENGTH. SUBTRACT
1374  REM  UNIT ATTRITION FROM
1375  REM  CURRENT UNIT STRENGTH
1376  REM  IF UNIT STRENGTH GOES
1377  REM  NEGATIVE, MAKE IT
1378  REM  EQUAL 0.
1379  REM
1380  FOR I = 1 TO M
1390 SB(I) = SB(I) - DB(I)
1400  IF SB(I) < 0 THEN SB(I) = 0
1410  IF SB(I) <  = BM * B1(I) THEN BR(I) = 8
1420  NEXT
1430  FOR J = 1 TO N
1440 SR(J) = SR(J) - DR(J)
1450  IF SR(J) < 0 THEN SR(J) = 0
1460  IF SR(J) <  = RM * R1(I) THEN SR(J) =  - 8
1470  NEXT
1480  RETURN
1490  REM  PRINTOUT/SUMMARY
1500  IF PR = 1 THEN  PR# 1
1510  PRINT T;: POKE 36,7: PRINT R0(KK);
1515  POKE 36,20: PRINT R0(KK)
1520  PR# 0
1530  RETURN
1540  REM  SAVE RESULTS IN A FILE
1550  PRINT
1555  INPUT "DO YOU WANT A FILE OF THIS RUN'S DATA? (Y/N) ";A$
1560  IF A$ = "N" THEN 1750
1570 D$ =  CHR$ (4)
1580  INPUT "NAME OF FILE FOR FORCE LEVEL DATA? ";F$
1590  PRINT D$;"OPEN ";F$;".COEFS"
1600  PRINT D$;"WRITE ";F$;".COEFS"
1610  PRINT A: PRINT B: PRINT P: PRINT D
1620  PRINT D$;"CLOSE ";F$;".COEFS"
1630  PRINT D$;"OPEN ";F$;".RESULTS"
1640  PRINT D$;"WRITE ";F$;".RESULTS"
1650  PRINT NK + 1: PRINT NK + 1
1660  FOR I = 0 TO NK
1670 T = I * 3
1680  PRINT T: PRINT B0(I)
1690  NEXT I
1700  FOR I = 0 TO NK
1710 T = I * 3
1720  PRINT T: PRINT R0(I)
1730  NEXT I
1740  PRINT D$;"CLOSE ";F$;".RESULTS"
1750  RETURN
```

```
1760  REM  PRINTOUT/SUMMARY
1770  HOME : VTAB 3
1780  IF PR = 1 THEN  PR# 1
1790  PRINT "TIME IS ";T
1800  PRINT : PRINT
1810  PRINT "BLUE FORCES"
1820  PRINT "UNIT"; TAB( 7);"XCOORD"; TAB( 15);"YCOORD"; TAB( 23);"SIZE"
1830  FOR I = 1 TO M
1840  PRINT " ";I; TAB( 7);XB(I); TAB( 15);YB(I); TAB( 24);SB(I)
1850  NEXT
1860  PRINT : PRINT "RED FORCES"
1870  PRINT "UNIT"; TAB( 7);"XCOORD"; TAB( 15);"YCOORD"; TAB( 23);"SIZE"
1880  FOR I = 1 TO N
1890  PRINT " ";I; TAB( 7);XR(I); TAB( 15);YR(I); TAB( 24);SR(I)
1900  NEXT
1910  PRINT : PRINT
1920  PR# 0
1930  INPUT "HIT RETURN TO CONTINUE";Z$
1940  RETURN
```

B. FORCE MAKER PROGRAM

```
10  REM  FORCE DATA-FILE MAKER
20  DIM BX(5,10),BY(5,10),B1(10)
30  GOTO 170
40  HOME
50  INPUT "NAME OF FILE? ";N$
60 D$ =  CHR$ (4)
70  PRINT D$;"OPEN ";N$;",";M$
80  PRINT D$;"WRITE ";N$;",";M$
90  PRINT M
100  FOR I = 1 TO M
110  PRINT B1(I)
120  FOR J = 1 TO B1(I)
130  PRINT BX(I,J): PRINT BY(I,J)
140  NEXT J
150  NEXT I
160  PRINT D$;"CLOSE ";N$;",";M$
170  HOME
180  VTAB 5
190  PRINT "MENU:"
200  PRINT "1 MAKE FILE FOR BLUE FORCES"
210  PRINT "2 MAKE FILE FOR RED FORCES"
220  PRINT "0 QUIT"
230  PRINT : INPUT "WHICH? ";Z
240  IF Z = 0 THEN  END
250  IF Z = 1 THEN M$ = "BLUE"
260  IF Z = 2 THEN M$ = "RED"
270  HOME
280  PRINT "WHAT IS THE FORCE SIZE OF ";M$;: INPUT " ";M
290  FOR I = 1 TO M
300  PRINT : PRINT "FOR UNIT ";I
310  PRINT "HOW MANY COORDINATING POINTS ";
315  INPUT "(INCLUDING START LOCATIONS)? ";B1(I)
320  FOR J = 1 TO B1(I)
330  PRINT
340  PRINT "ENTER GRID COORDS FOR COORDINATING POINT # ";J
350  INPUT "X-COORD IS ";BX(I,J)
360  INPUT "Y-COORD IS ";BY(I,J)
370  NEXT J
380  NEXT I
```

```
390  HOME
400  PRINT : PRINT  TAB( 12);"*** DATA CHECK ***"
410  PRINT : PRINT : PRINT
420  PRINT "UNIT PT"; TAB( 9);"X-COORD"; TAB( 17);"Y-COORD"
430  FOR I = 1 TO M
440  FOR J = 1 TO B1(I)
450  PRINT  TAB( 2);I; TAB( 6);J; TAB( 10);BX(I,J); TAB( 18);BY(I,J)
460  NEXT J
470  NEXT I
480  INPUT "ANY CHANGES? (Y/N) ";Z$
490  IF Z$ = "N" THEN 40
500  PRINT
510  INPUT "WHICH UNIT? ";I
520  INPUT "WHICH POINT? ";J
530  INPUT "X-COORD IS ";BX(I,J)
540  INPUT "Y-COORD IS ";BY(I,J)
550  GOTO 390
```

C. RESULTS READER PROGRAM

```
10  HOME : VTAB 3
20 D$ =  CHR$ (4)
30  INPUT "NAME OF FILE FOR FORCE LEVEL DATA? ";F$
40  PRINT D$;"OPEN ";F$;".COEFS"
50  PRINT D$;"READ ";F$;".COEFS"
60  INPUT A: INPUT B: INPUT P: INPUT Q
70  PRINT D$;"CLOSE ";F$;".COEFS"
80  PRINT D$;"OPEN ";F$;".RESULTS"
90  PRINT D$;"READ ";F$;".RESULTS"
100  INPUT N: INPUT N
110 NK = N - 1
120  DIM B0(NK),R0(NK)
130  FOR I = 0 TO NK
140  INPUT T: INPUT B0(I)
150  NEXT I
160  FOR I = 0 TO NK
170  INPUT T: INPUT R0(I)
180  NEXT I
190  PRINT D$;"CLOSE ";F$;".RESULTS"
200  INPUT "PRINTER ON? ";A$
210  HOME : VTAB 3
220  IF A$ = "Y" THEN  PR# 1
230  PRINT "A = ";A
240  PRINT "B = ";B
250  PRINT "P = ";P
260  PRINT "Q = ";Q
270  PRINT
280  PRINT "TIME":: POKE 36,7: PRINT "BLUE";
285  POKE 36,20: PRINT "RED"
290  FOR KK = 0 TO NK
300 T = 30 * KK
310  PRINT T;: POKE 36,7: PRINT B0(KK);
315  POKE 36,20: PRINT R0(KK)
320  NEXT KK
330  PR# 0
340  END
```

D. MAP MAKER PROGRAM

```
10  REM   MAP CATALOG MAKER/READER
20  DIM HL(10,10,12)
30  HOME : VTAB 3
40  PRINT "MENU:"
50  PRINT "1 MAP MAKER"
60  PRINT "2 MAP READER"
70  PRINT "0 QUIT"
80  PRINT : INPUT "WHICH? ";X
90  IF X = 0 THEN  END
100  ON X GOTO 120,800
110  REM   MAP CATALOG MAKER
120  HOME : VTAB 2
130  PRINT  TAB( 11);"*** MAP MAKER ***"
140  PRINT : PRINT
150  INPUT "NEW OR OLD MAP? (N/O) ";Z$
160  IF Z$ = "N" THEN II = 1:JJ = 1: GOTO 220
170  INPUT "NAME OF MAP? ";M$
180 D$ =  CHR$ (4)
190  GOSUB 620
200  INPUT "ENTER STARTING GRID SQUARE (I,J) ";II,JJ
210  GOTO 250
220  INPUT "SIZE OF MAP (1000 M SQUARES)? ";L,W
230  INPUT "ENTER LOWER LEFT GRID SQUARE (EG. 50,33) ";LX,LY
240  INPUT "BASE ELEVATION OF MAP? ";HL(0,0,0)
250  FOR I = II TO L
260  FOR J = JJ TO W
270  PRINT : PRINT
280  PRINT "FOR GRID SQUARE ";I;".";J
290  INPUT "NUMBER OF HILL MASSES? ";HL(I,J,0):KK = HL(I,J,0)
300  IF KK = 0 THEN 380
310  PRINT "ENTER THE ";KK;" HILLS FOR GRID SQUARE ";I;",";J
320  FOR K = 1 TO KK
330  PRINT "HILL ";K;" IS ";: INPUT HL(I,J,K)
340  NEXT K
350 JJ = 1
360  PRINT : INPUT "SAVE? (Y/N)";A$
370  IF A$ = "Y" THEN 400
380  NEXT J
390  NEXT I
```

```
400  HOME : UTAB 4
410  INPUT "NAME OF MAP FILE? ";M$
420 D$ =  CHR$ (4)
430  PRINT D$;"OPEN ";M$;".MAP"
440  PRINT D$;"WRITE ";M$;".MAP"
450  PRINT L: PRINT W
460  PRINT LX: PRINT LY
470  PRINT HL(0,0,0)
480  FOR I = 1 TO L
490  FOR J = 1 TO W
500  PRINT HL(I,J,0)
510  IF HL(I,J,0) = 0 THEN 550
520  FOR K = 1 TO HL(I,J,0)
530  PRINT HL(I,J,K)
540  NEXT K
550  NEXT J
560  NEXT I
570  PRINT D$;"CLOSE ";M$;".MAP"
580  PRINT : PRINT : PRINT : PRINT "... DONE": PRINT : PRINT
590  INPUT "DO YOU WANT TO CONTINUE INPUTTING DATA? (Y/N) ";Z$
600  IF Z$ = "Y" THEN 200
610  GOTO 30
620 D$ =  CHR$ (4)
630  PRINT D$;"OPEN ";M$;".MAP"
640  PRINT D$;"READ ";M$;".MAP"
650  INPUT L: INPUT W
660  INPUT LX: INPUT LY
670  INPUT HL(0,0,0)
680  FOR I = 1 TO L
690  FOR J = 1 TO W
700  INPUT HL(I,J,0)
710  IF HL(I,J,0) = 0 THEN 750
720  FOR K = 1 TO HL(I,J,0)
730  INPUT HL(I,J,K)
740  NEXT K
750  NEXT J
760  NEXT I
770  PRINT D$;"CLOSE ";M$;".MAP"
780  RETURN
790  REM  MAP CATALOG FILE READER
800  HOME : UTAB 2
810  PRINT  TAB( 11);"*** MAP READER ***"
820  PRINT : PRINT
```

```
830  INPUT "NAME OF MAP FILE? ";M$
840  INPUT "PRINTER ON? (Y/N) ";A$
850 D$ =  CHR$ (4)
860  PRINT D$;"OPEN ";M$;".MAP"
870  PRINT D$;"READ ";M$;".MAP"
880  INPUT L: INPUT W
890  INPUT LX: INPUT LY
900  INPUT HL(0,0,0)
910  FOR I = 1 TO L
920  FOR J = 1 TO W
930  INPUT HL(I,J,0)
940  IF HL(I,J,0) = 0 THEN 980
950  FOR K = 1 TO HL(I,J,0)
960  INPUT HL(I,J,K)
970  NEXT K
980  NEXT J
990  NEXT I
1000  PRINT D$;"CLOSE ";M$;".MAP"
1010  HOME
1020  IF A$ = "Y" THEN  PR# 1
1030  PRINT "NAME OF MAP: ";M$
1040  PRINT "LOWER LEFT GRID SQUARE: ";LX;" ";LY
1050  PRINT "SIZE OF MAP: ";L;" BY ";W
1060  PRINT "BASE ELEVATION: ";HL(0,0,0)
1070  PRINT
1080  FOR I = 1 TO L
1090  FOR J = 1 TO W
1100  PRINT I,J,HL(I,J,0)
1110  IF HL(I,J,0) = 0 THEN 1160
1120  FOR K = 1 TO HL(I,J,0)
1130  PRINT HL(I,J,K);" ";
1140  NEXT K
1150  PRINT
1160  NEXT J
1170  PRINT : NEXT I
1180  PR# 0
1190  PRINT : PRINT
1200  INPUT "HIT RETURN TO CONTINUE";X$
1210  GOTO 30
```

E. HILL MAKER PROGRAM

```
10  REM  -  HILL DATA FILE MAKER/READER
20  DIM H(200,8),K(8)
30 K(1) = 5:K(2) = 9:K(3) = 14:K(4) = 18
40 K(5) = 22:K(6) = 24:K(7) = 29:K(8) = 33
50  REM      HILL DATA FILE MAKER
60  HOME : VTAB 2
70  PRINT "MENU:"
80  PRINT "1 HILL MAKER"
90  PRINT "2 HILL READER"
100  PRINT "0 QUIT"
110  PRINT : INPUT "WHICH? ";X
120  IF X = 0 THEN  END
130  ON X GOTO 150,820
140  REM     HILL DATA FILE MAKER
150  HOME : VTAB 2
160  PRINT  TAB( 11);"*** HILL MAKER ***"
170  PRINT : PRINT
180  INPUT "NEW OR OLD FILE? (N/O) ";A$
190  IF A$ = "N" THEN 230
200  GOSUB 560
210  INPUT "WHICH HILL DO YOU WANT TO START WITH? ";II
220  GOTO 260
230  HOME : VTAB 4
240 II = 1
250  INPUT "NUMBER OF HILLS? ";N
260  FOR I = II TO N
270  PRINT "ENTER THE DATA FOR HILL ";I
280  INPUT "XCOORD IS ";H(I,1)
290  INPUT "YCOORD IS ";H(I,2)
300  INPUT "PEAK ELEVATION IS ";H(I,3)
310  INPUT "ORIENTATION ANGLE IS ";H(I,4)
320  INPUT "ECCENTRICITY IS ";H(I,5)
330  INPUT "SPREAD IS ";H(I,6)
340  INPUT "MAX HEIGHT IS ";H(I,7)
350  INPUT "CUT HEIGHT IS ";H(I,8)
360  PRINT : PRINT : INPUT "SAVE? (Y/N) ";Z$
370  IF Z$ = "Y" THEN 390
380  NEXT I
390  HOME : VTAB 4
400 D$ =  CHR$ (4)
410  INPUT "NAME OF FILE? ";N$
420  PRINT D$;"OPEN ";N$;".HILLS, L34"
430  PRINT D$;" WRITE ";N$;".HILLS, R0"
440  PRINT N
450  FOR I = 1 TO N
460  PRINT D$;"WRITE ";N$;".HILLS, R";I
```

```
470  FOR J = 1 TO 8
480  PRINT H(I,J)
490  NEXT J
500  NEXT I
510  PRINT D$;"CLOSE ";N$;".HILLS"
520  PRINT : PRINT : PRINT "... DONE"
525  PRINT : PRINT
530  INPUT "DO YOU WANT TO CONTINUE INPUTTING DATA? (Y/N) ";A$
540  IF A$ = "Y" THEN 210
550  GOTO 60
560  HOME
570 D$ =  CHR$ (4)
580  INPUT "NAME OF FILE? ";N$
590  PRINT D$;"OPEN ";N$;".HILLS. L34"
600  PRINT D$;"READ ";N$;".HILLS, R0"
610  INPUT N
620  FOR I = 1 TO N
630  PRINT D$;"READ ";N$;".HILLS, R";I
640  FOR J = 1 TO 8
650  INPUT H(I,J)
660  NEXT J
670  NEXT I
680  PRINT D$;"CLOSE ";N$;".HILLS"
690  HOME
700  PRINT "NUMBER OF HILLS: ";N: PRINT
710 M = 1
720  FOR I = 1 TO N
730  IF M = 11 THEN  PRINT :M = 1
740  PRINT I;
750  FOR J = 1 TO 8
760  PRINT  TAB( K(J));H(I,J);
770  NEXT J
780  PRINT
790 M = M + 1
800  NEXT I
810  RETURN
```

```
820  REM  HILL DATA FILE READER
830 PR = 0
840  HOME : VTAB 2
850  PRINT  TAB( 10);"*** HILL READER ***"
860  PRINT : PRINT
870 D$ =  CHR$ (4)
880  INPUT "NAME OF MAP? ";N$
890  INPUT "PRINTER ON? (Y/N) ";Z$
900  IF Z$ = "Y" THEN PR = 1
910  PRINT D$;"OPEN ";N$;".HILLS, L34"
920  PRINT D$;"READ ";N$;".HILLS, R0"
930  INPUT N
940  FOR I = 1 TO N
950  PRINT D$;"READ ";N$;".HILLS, R";I
960  FOR J = 1 TO 3
970  INPUT H(I,J)
980  NEXT J
990  NEXT I
1000  PRINT D$;"CLOSE ";N$;".HILLS"
1010  HOME
1020  IF PR = 1 THEN  PR# 1
1030  PRINT : PRINT N$;" HILL DATA": PRINT
1040  PRINT "NUMBER OF HILLS: ";N
1050  PRINT
1060 M = 1
1070  FOR I = 1 TO N
1080  IF M = 11 THEN  PRINT :M = 1
1090  PRINT I;
1100  FOR J = 1 TO 3
1110 KK = K(J) + 3 * J
1120  IF PR = 1 THEN  POKE 36,KK: PRINT H(I,J);: GOTO 1140
1130  PRINT  TAB( K(J));H(I,J);
1140  NEXT J
1150  PRINT
1160 M = M + 1
1170  NEXT I
1180  PR# 0
1190  INPUT "HIT RETURN TO CONTINUE";X$
1200  GOTO 60
```

F. WOODS MAKER PROGRAM

```
10  REM   WOODS CATALOG MAKER/READER
20  DIM FL(10,10,9)
30  HOME : VTAB 3
40  PRINT "MENU:"
50  PRINT "1 WOODS MAKER"
60  PRINT "2 WOODS READER"
70  PRINT "0 QUIT"
80  PRINT : INPUT "WHICH? ";X
90  IF X = 0 THEN  END
100  ON X GOTO 120,770
110  REM  WOODS CATALOG MAKER
120  HOME : VTAB 2
130  PRINT  TAB( 11);"*** WOODS MAKER ***"
140  PRINT : PRINT
150  INPUT "NEW OR OLD MAP? (N/O) ";Z$
160  IF Z$ = "N" THEN II = 1:JJ = 1: GOTO 220
170  INPUT "NAME OF MAP? ";M$
180 D$ =  CHR$ (4)
190  GOSUB 800
200  INPUT "ENTER STARTING GRID SQUARE (I,J) ";II,JJ
210  GOTO 240
220  INPUT "SIZE OF MAP (1000 M SQUARES)? ";L,W
230  INPUT "ENTER LOWER LEFT GRID SQUARE (EG. 50,93) ";LX,LY
240  FOR I = II TO L
250  FOR J = JJ TO W
260  PRINT : PRINT
270  PRINT "FOR GRID SQUARE ";I;",";J
280  INPUT "NUMBER OF FORESTS? ";FL(I,J,0):KK = FL(I,J,0)
290  IF KK = 0 THEN 370
300  PRINT "ENTER THE ";KK;" FORESTS FOR GRID SQUARE ";I;",";J
310  FOR K = 1 TO KK
320  PRINT "FOREST ";K;" IS ";: INPUT FL(I,J,K)
330  NEXT K
340 JJ = 1
350  PRINT : INPUT "SAVE? (Y/N)";A$
360  IF A$ = "Y" THEN 390
370  NEXT J
380  NEXT I
```

```
390  HOME : VTAB 4
400  INPUT "NAME OF MAP FILE? ";M$
410 D$ =  CHR$ (4)
420  PRINT D$;"OPEN ";M$;".WOODS"
430  PRINT D$;"WRITE ";M$;".WOODS"
440  PRINT L: PRINT W
450  PRINT LX: PRINT LY
460  FOR I = 1 TO L
470  FOR J = 1 TO W
480  PRINT FL(I,J,0)
490  IF FL(I,J,0) = 0 THEN 530
500  FOR K = 1 TO FL(I,J,0)
510  PRINT FL(I,J,K)
520  NEXT K
530  NEXT J
540  NEXT I
550  PRINT D$;"CLOSE ";M$;".WOODS"
560  PRINT : PRINT : PRINT : PRINT "... DONE": PRINT : PRINT
570  INPUT "DO YOU WANT TO CONTINUE INPUTTING DATA? (Y/N) ";Z$
580  IF Z$ = "Y" THEN 200
590  GOTO 30
600 D$ =  CHR$ (4)
610  PRINT D$;"OPEN ";M$;".WOODS"
620  PRINT D$;"READ ";M$;".WOODS"
630  INPUT L: INPUT W
640  INPUT LX: INPUT LY
650  FOR I = 1 TO L
660  FOR J = 1 TO W
670  INPUT FL(I,J,0)
680  IF FL(I,J,0) = 0 THEN 720
690  FOR K = 1 TO FL(I,J,0)
700  INPUT FL(I,J,K)
710  NEXT K
720  NEXT J
730  NEXT I
740  PRINT D$;"CLOSE ";M$;".WOODS"
750  RETURN
```

```
760  REM WOODS CATALOG FILE READER
770  HOME : VTAB 2
780  PRINT  TAB( 11);"*** WOODS READER ***"
790  PRINT : PRINT
800  INPUT "NAME OF MAP FILE? ";M$
810  INPUT "PRINTER ON? (Y/N) ";A$
820 D$ =  CHR$ (4)
830  PRINT D$;"OPEN ";M$;".WOODS"
840  PRINT D$;"READ ";M$;".WOODS"
850  INPUT L: INPUT W
860  INPUT LX: INPUT LY
870  FOR I = 1 TO L
880  FOR J = 1 TO W
890  INPUT FL(I,J,0)
900  IF FL(I,J,0) = 0 THEN 940
910  FOR K = 1 TO FL(I,J,0)
920  INPUT FL(I,J,K)
930  NEXT K
940  NEXT J
950  NEXT I
960  PRINT D$;"CLOSE ";M$;".WOODS"
970  HOME
980  IF A$ = "Y" THEN  PR# 1
990  PRINT "NAME OF MAP: ";M$
1000  PRINT "LOWER LEFT GRID SQUARE: ";LX;" ";LY
1010  PRINT "SIZE OF MAP: ";L;" BY ";W
1020  PRINT
1030  FOR I = 1 TO L
1040  FOR J = 1 TO W
1050  PRINT I,J,FL(I,J,0)
1060  IF FL(I,J,0) = 0 THEN 1110
1070  FOR K = 1 TO FL(I,J,0)
1080  PRINT FL(I,J,K);" ";
1090  NEXT K
1100  PRINT
1110  NEXT J
1120  PRINT : NEXT I
1130  PR# 0
1140  PRINT : PRINT
1150  INPUT "HIT RETURN TO CONTINUE";X$
1160  GOTO 30
```

G. FOREST MAKER PROGRAM

```
10  REM  FOREST MAKER/READER
20  DIM F(120,6),K(6)
30 K(1) = 5:K(2) = 12:K(3) = 20:K(4) = 25
40 K(5) = 30:K(6) = 36
50  HOME : VTAB 2
60  PRINT "MENU: "
70  PRINT "1 FOREST MAKER"
80  PRINT "2 FOREST READER"
90  PRINT "0 QUIT"
100  PRINT : INPUT "WHICH? ";X
110  IF X = 0 THEN  END
120  ON X GOTO 140,300
130  REM   FOREST DATA FILE MAKER
140  HOME : VTAB 2
150  PRINT  TAB( 10);"*** FOREST MAKER ***"
160  PRINT : PRINT
170  INPUT "NEW OR OLD FILE? (N/O) ";A$
180  IF A$ = "N" THEN 220
190  GOSUB 530
200  INPUT "WHICH FOREST DO YOU WANT TO START WITH? ";II
210  GOTO 250
220  HOME : VTAB 4
230 II = 1
240  INPUT "NUMBER OF FORESTS? ";N
250  FOR I = II TO N
260  PRINT "ENTER THE DATA FOR FOREST ";I
270  INPUT "XCOORD IS ";F(I,1)
280  INPUT "YCOORD IS ";F(I,2)
290  INPUT "HEIGHT OF TREES IS ";F(I,3)
300  INPUT "ORIENTATION ANGLE IS ";F(I,4)
310  INPUT "SEMI-MAJOR AXIS LENGTH IS ";F(I,5)
320  INPUT "SEMI-MINOR AXIS LENGTH IS ";F(I,6)
330  PRINT : PRINT : INPUT "SAVE? (Y/N) ";Z$
340  IF Z$ = "Y" THEN 360
350  NEXT I
360  HOME : VTAB 4
370 D$ =  CHR$ (4)
```

```
380  INPUT "NAME OF FILE? ";N$
390  PRINT D$;"OPEN ";N$;",FORESTS, L38"
400  PRINT D$;" WRITE ";N$;",FORESTS, R0"
410  PRINT N
420  FOR I = 1 TO N
430  PRINT D$;"WRITE ";N$;",FORESTS, R";I
440  FOR J = 1 TO 6
450  PRINT F(I,J)
460  NEXT J
470  NEXT I
480  PRINT D$;"CLOSE ";N$;",FORESTS"
490  PRINT : PRINT : PRINT "... DONE"
495  PRINT : PRINT
500  INPUT "DO YOU WANT TO CONTINUE INPUTTING DATA? (Y/N) ";A$
510  IF A$ = "Y" THEN 200
520  GOTO 50
530  HOME
540 D$ =  CHR$ (4)
550  INPUT "NAME OF FILE? ";N$
560  PRINT D$;"OPEN ";N$;",FORESTS, L38"
570  PRINT D$;"READ ";N$;",FORESTS, R0"
580  INPUT N
590  FOR I = 1 TO N
600  PRINT D$;"READ ";N$;",FORESTS, R";I
610  FOR J = 1 TO 6
620  INPUT F(I,J)
630  NEXT J
640  NEXT I
650  PRINT D$;"CLOSE ";N$;",FORESTS"
660  HOME
670  PRINT N$;" FORESTS LIST": PRINT
680  PRINT "NUMBER OF FORESTS: ";N: PRINT
690 M = 1
700  FOR I = 1 TO N
710  IF M = 11 THEN  PRINT :M = 1
720  PRINT I;
730  FOR J = 1 TO 6
740  PRINT  TAB( K(J));F(I,J);
750  NEXT J
760  PRINT
770 M = M + 1
780  NEXT I
790  RETURN
```

```
800  REM   FOREST DATA FILE READER
810  HOME : VTAB 2
820 PR = 0
830  PRINT  TAB( 10);"*** FOREST READER ***"
840  PRINT : PRINT
850 D$ =  CHR$ (4)
860  INPUT "NAME OF MAP? ";N$
870  INPUT "PRINTER ON? (Y/N) ";Z$
880  IF Z$ = "Y" THEN PR = 1
890  PRINT D$;"OPEN ";N$;".FORESTS, L39"
900  PRINT D$;"READ ";N$;".FORESTS, R0"
910  INPUT N
920  FOR I = 1 TO N
930  PRINT D$;"READ ";N$;".FORESTS, R";I
940  FOR J = 1 TO 6
950  INPUT F(I,J)
960  NEXT J
970  NEXT I
980  PRINT D$;"CLOSE ";N$;".FORESTS"
990  HOME
1000  IF PR = 1 THEN  PR# 1
1010  PRINT : PRINT N$;" FOREST DATA": PRINT
1020  PRINT "NUMBER OF FORESTS ";N
1030  PRINT
1040 M = 1
1050  FOR I = 1 TO N
1060  IF M = 11 THEN  PRINT :M = 1
1070  PRINT I;
1080  FOR J = 1 TO 6
1090 KK = K(J) + 3 * J
1100  IF PR = 1 THEN  POKE 36,KK: PRINT F(I,J);: GOTO 1120
1110  PRINT  TAB( K(J));F(I,J);
1120  NEXT J
1130  PRINT
1140 M = M + 1
1150  NEXT I
1160  PR# 0
1170  INPUT "HIT RETURN TO CONTINUE";X$
1180  GOTO 50
```

H. TERRAIN MODEL PROGRAM

```
10  REM  UPDATED 02 FEB 82
20  REM  DAMSTAC TERRAIN MODEL
30  DIM XB(5),YB(5),XR(5),YR(5),RX(5,10),BY(5,10),RX(5,10)
35  DIM ZB(5),ZR(5),HL(10,10,12)
40  DIM BP(5),RP(5),SB(5),SR(5),RY(5,10)
50  DIM BF(5),RF(5),BT(5,5),RT(5,5),OB(5),OR(5)
60  DIM BR(5),RR(5)
70  DIM B1(5),R1(5)
80 T = 0:OT = 30:MT = 1500
90 D$ =  CHR$ (4)
100 P = 0.99:Q = 0.93
110 PI = 3.141592654:PR = 0
120 A = 2.4:B = 0.03
130 BE = 0.3:RE = 0.3
140 BM = .5:RM = .5
150 KK = 0
160 NK = MT / OT
165 ZZ = 1
170  DIM B0(NK),R0(NK)
171  HOME
172  INPUT "NAME OF MAP FILE? ";M$
173  PRINT D$;"OPEN ";M$;".MAP"
174  PRINT D$;"READ ";M$;".MAP"
175  INPUT L: INPUT W
176  INPUT LX: INPUT LY
177  INPUT HL(0,0,0)
178  FOR I = 1 TO L
179  FOR J = 1 TO W
180  INPUT HL(I,J,0)
181  IF HL(I,J,0) = 0 THEN 184
182  FOR K = 1 TO HL(I,J,0)
183  INPUT HL(I,J,K)
184  NEXT : NEXT : NEXT
185  PRINT D$;"CLOSE ";M$;".MAP"
186 LX = LX - 1:LY = LY - 1
187  PRINT D$;"OPEN ";M$;".HILLS, L34"
188  PRINT D$;"READ ";M$;".HILLS, R0"
189  INPUT NH: DIM H(NH,8)
190  FOR I = 1 TO NH
191  PRINT D$;"READ ";M$;".HILLS, R";I
192  FOR J = 1 TO 8
193  INPUT H(I,J)
194  NEXT : NEXT
195  PRINT D$;"CLOSE ";M$;".HILLS"
```

```
198  HOME
199  INPUT "NAME OF FILE FOR BLUE DATA? ";N$
200  PRINT D$;"OPEN ";N$;".BLUE"
210  PRINT D$;"READ ";N$;".BLUE"
220  INPUT M
230  FOR I = 1 TO M
240  INPUT B1
250  FOR J = 1 TO B1
260  INPUT BX(I,J): INPUT BY(I,J)
270  NEXT : NEXT
280  PRINT D$;"CLOSE ";N$;".BLUE"
290  FOR I = 1 TO M:XB(I) = BX(I,1):YB(I) = BY(I,1):BP(I) = 1
300  PRINT B1: FOR JJ = 1 TO B1: PRINT BX(I,JJ),BY(I,JJ): NEXT JJ
310  PRINT "SIZE OF BLUE UNIT ";I;: INPUT SB(I)
320 B1(I) = SB(I)
330 B0(0) = B0(0) + SB(I)
340 BR(I) = 0
341 X = XB(I):Y = YB(I)
342 X4 =  INT (XB(I) / 1000) - LX:Y4 =  INT (YB(I) / 1000) - LY
343  GOSUB 1960
344 ZB(I) = Z
345  PRINT "ELEVATION OF BLUE ";I;" IS ";Z
346  FOR I1 = 1 TO HL(X4,Y4,0):H2 = HL(X4,Y4,I1):H(H2,0) = 0: NEXT
350  NEXT
360  PRINT
370  INPUT "NAME OF FILE FOR RED DATA? ";N$
380  PRINT D$;"OPEN ";N$;".RED"
390  PRINT D$;"READ ";N$;".RED"
400  INPUT N
410  FOR I = 1 TO N
420  INPUT R1
430  FOR J = 1 TO R1
440  INPUT RX(I,J): INPUT RY(I,J)
450  NEXT : NEXT
460  PRINT D$;"CLOSE ";N$;".RED"
470  FOR I = 1 TO N:XR(I) = RX(I,1):YR(I) = RY(I,1):RP(I) = 1
480  PRINT "SIZE OF RED UNIT ";I;: INPUT SR(I)
490 R1(I) = SR(I)
500 R0(0) = R0(0) + SR(I)
510 RR(I) = 5
511 X = XR(I):Y = YR(I)
512 X4 =  INT (XR(I) / 1000) - LX:Y4 =  INT (YR(I) / 1000) - LY
513  GOSUB 1960
514 ZR(I) = Z
515  PRINT "ELEVATION OF RED ";I;" IS ";Z
516  FOR I1 = 1 TO HL(X4,Y4,0):H2 = HL(X4,Y4,I1):H(H2,0) = 0: NEXT
520  NEXT I
```

```
530  PRINT : INPUT "DO YOU WANT THE PRINTER ON? (Y/N) ";A$
535  IF A$ = "Y" THEN PR = 1
540  HOME : UTAB 3
550  IF PR = 1 THEN  PR# 1
560  PRINT
570  PRINT "TIME";: POKE 36,7: PRINT "BLUE";: POKE 36,20: PRINT "RED"
580  PR# 0
590  IF ZZ = 1 THEN  GOSUB 1760: GOTO 610
600  GOSUB 1490
610  REM  MAIN PROGRAM
620 T = T + DT
630  IF T <  = MT THEN 640
635  PRINT "END OF BATTLE DUE TO TIME": GOSUB 1540: END
640  GOSUB 790
650  GOSUB 1170
660 KK = KK + 1
670 B0(KK) = 0:R0(KK) = 0
680  FOR I = 1 TO M
690 B0(KK) = B0(KK) + SB(I)
700  NEXT I
710  FOR J = 1 TO N
720 R0(KK) = R0(KK) + SR(J)
730  NEXT J
740  IF ZZ = 1 THEN  GOSUB 1760: GOTO 760
750  GOSUB 1490
760  IF B0(KK) > BE * B0(0) THEN 770
765  PRINT : PRINT "END OF BATTLE DUE TO ATTRITION OF BLUE"
766  GOSUB 1540: END
770  IF R0(KK) > RE * R0(0) THEN 780
775  PRINT : PRINT "END OF BATTLE DUE TO ATTRITION OF RED"
776  GOSUB 1540: END
780  GOTO 620
790  REM  MOVEMENT SUBROUTINE
800  FOR I = 1 TO N
810 J = RP(I)
820 DY = RY(I,J + 1) - YR(I)
830 DX = RX(I,J + 1) - XR(I)
840  IF DX = 0 AND DY > 0 THEN AN = PI / 2: GOTO 880
850  IF DX = 0 AND DY < 0 THEN AN = 3 * PI / 2: GOTO 880
860 AN =  ATN (DY / DX)
870  IF DX < 0 THEN AN = PI + AN
880 D1 = DT * RR(I) *  COS (AN)
890 D2 = DT * RR(I) *  SIN (AN)
900 DH =  SQR (D1 ^ 2 + D2 ^ 2)
910 DI =  SQR (DX ^ 2 + DY ^ 2)
920  IF DI >  = DH THEN 950
930 D1 = DX:D2 = DY
940 RP(I) = J + 1
```

```
950 YR(I) =  INT (YR(I) + D2)
960 XR(I) =  INT (XR(I) + D1)
961 X = XR(I):Y = YR(I)
962 X4 =  INT (XR(I) / 1000) - LX:Y4 =  INT (YR(I) / 1000) - LY
963  GOSUB 1960
964 ZR(I) = Z
965  PRINT "ELEVATION OF RED ";I;" IS ";Z
966  FOR I1 = 1 TO HL(X4,Y4,0):H2 = HL(X4,Y4,I1):H(H2,0) = 0: NEXT
970  NEXT
980  FOR I = 1 TO M
990 J = BP(I)
1000 DY = BY(I,J + 1) - YB(I)
1010 DX = BX(I,J + 1) - XB(I)
1020  IF DX = 0 AND DY > 0 THEN AN = PI / 2: GOTO 1060
1030  IF DX = 0 AND DY < 0 THEN AN = 3 * PI / 2: GOTO 1060
1040 AN =  ATN (DY / DX)
1050  IF DX < 0 THEN AN = PI + AN
1060 D1 = DT * BR(I) *  COS (AN)
1070 D2 = DT * BR(I) *  SIN (AN)
1080 DH =  SQR (D1 ^ 2 + D2 ^ 2)
1090 DI =  SQR (DX ^ 2 + DY ^ 2)
1100  IF DI >  = DH THEN 1130
1110 D1 = DX:D2 = DY
1120 BP(I) = J + 1
1130 YB(I) =  INT (YB(I) + D2)
1140 XB(I) =  INT (XB(I) + D1)
1150  NEXT
1160  RETURN
1170  REM   ATTRITION SUBROUTINE
1180  FOR I = 1 TO M:BF(I) = 0:DB(I) = 0: NEXT
1190  FOR J = 1 TO N:RF(J) = 0:DR(J) = 0: NEXT
1200  FOR I = 1 TO M
1210  FOR J = 1 TO N
1220 BT(I,J) = 0:RT(J,I) = 0
1230 RG =  SQR ((XB(I) - XR(J)) ^ 2 + (YB(I) - YR(J)) ^ 2)
1240  IF RG > 2700 THEN 1310
1250 RR(I) = 2
1251  REM  CHECK LINE OF SIGHT
1252 X1 =  INT (XB(I) / 1000) - LX:Y1 =  INT (YB(I) / 1000) - LY
1253 X2 =  INT (XR(J) / 1000) - LX:Y2 =  INT (YR(J) / 1000) - LY
1254 X3 = X2 - X1:Y3 = Y2 - Y1
1255  FOR I1 = 0 TO  ABS (X3)
1256 X4 = X1 + I1 *  SGN (X3)
1257  FOR J1 = 0 TO  ABS (Y3)
1258 Y4 = Y1 + J1 *  SGN (Y3)
1261  PRINT "CHECKING GRID SQUARE ";X4;",";Y4;" FOR BLUE ";I;" RED ";J
```

```
1262  GOSUB 1960
1263  NEXT : NEXT
1264  FOR I1 = 1 TO NH:H(I1,0) = 0: NEXT
1265  REM   LOS IS OK
1270 BF(I) = BF(I) + 1
1280 RF(J) = RF(J) + 1
1290 BT(I,J) = A * (1 - (P) ^ SB(I)) * SR(J)
1300 RT(J,I) = B * (1 - (Q) ^ SR(J)) * SB(I)
1310  NEXT : NEXT
1320  FOR I = 1 TO M
1330  FOR J = 1 TO N
1340  IF RF(J) = 0 OR BF(I) = 0 THEN 1370
1350 OB(I) = OB(I) + BT(I,J) / RF(J)
1360 OR(J) = OR(J) + RT(J,I) / BF(I)
1370  NEXT : NEXT
1380  FOR I = 1 TO M
1390 SB(I) = SB(I) - OB(I)
1400  IF SB(I) < 0 THEN SB(I) = 0
1410  IF SB(I) <  = BM * B1(I) THEN BR(I) = 8
1420  NEXT
1430  FOR J = 1 TO N
1440 SR(J) = SR(J) - OR(J)
1450  IF SR(J) < 0 THEN SR(J) = 0
1460  IF SR(I) <  = RM * R1(I) THEN RR(I) =  - 8
1470  NEXT
1480  RETURN
1490  REM  PRINTOUT/SUMMARY
1500  IF PR = 1 THEN  PR# 1
1510  PRINT T;: POKE 36,7: PRINT B0(KK);: POKE 36,20: PRINT R0(KK)
1520  PR# 0
1530  RETURN
1540  REM  SAVE RESULTS IN A FILE
1550  PRINT : INPUT "DO YOU WANT A FILE OF THIS RUN'S DATA? (Y/N) ";A$
1560  IF A$ = "N" THEN 1750
1570 D$ =  CHR$ (4)
1580  INPUT "NAME OF FILE FOR FORCE LEVEL DATA? ";F$
1590  PRINT D$;"OPEN ";F$;".COEFS"
1600  PRINT D$;"WRITE ";F$;".COEFS"
1610  PRINT A: PRINT B: PRINT P: PRINT Q
1620  PRINT D$;"CLOSE ";F$;".COEFS"
1630  PRINT D$;"OPEN ";F$;".RESULTS"
1640  PRINT D$;"WRITE ";F$;".RESULTS"
1650  PRINT NK + 1: PRINT NK + 1
1660  FOR I = 0 TO NK
1670 T = I * 3
1680  PRINT T: PRINT B0(I)
1690  NEXT I
```

```
1700  FOR I = 0 TO NK
1710 T = I * 3
1720  PRINT T: PRINT R0(I)
1730  NEXT I
1740  PRINT D$;"CLOSE ";F$;".RESULTS"
1750  RETURN
1760  REM  PRINTOUT/SUMMARY
1770  HOME : UTAB 3
1780  IF PR = 1 THEN  PR# 1
1790  PRINT "TIME IS ";T
1800  PRINT : PRINT
1810  PRINT "BLUE FORCES"
1820  PRINT "UNIT"; TAB( 7);"XCOORD"; TAB( 15);"YCOORD"; TAB( 23);"SIZE"
1830  FOR I = 1 TO M
1840  PRINT " ";I; TAB( 7);XB(I); TAB( 15);YB(I); TAB( 24);SB(I)
1850  NEXT
1860  PRINT : PRINT "RED FORCES"
1870  PRINT "UNIT"; TAB( 7);"XCOORD"; TAB( 15);"YCOORD"; TAB( 23);"SIZE"
1880  FOR I = 1 TO N
1890  PRINT " ";I; TAB( 7);XR(I); TAB( 15);YR(I); TAB( 24);SR(I)
1900  NEXT
1910  PRINT : PRINT
1920  PR# 0
1930  INPUT "HIT RETURN TO CONTINUE";Z$
1940  RETURN
1950  REM  ELEVATION COMPUTATION
1960 CR =  - 99999999
1970 Z = HL(0,0,0)
1980  FOR H1 = 1 TO HL(X4,Y4,0)
1990 H2 = HL(X4,Y4,H1)
2000  IF H(H2,0) = 1 THEN 2150
2010 XS = X - H(H2,1) * 100:YS = Y - H(H2,2) * 100
2020 A1 =  LOG (H(H2,7) / (H(H2,7) - 50))
2030 B1 = A1 * (H(H2,5)) ^ 2
2040 C1 = H(H2,4) * PI / 180
2050 C2 = H(H2,6) ^ 2
2060 P1 =  - (A1 * ( COS (C1)) ^ 2 + B1 * ( SIN (C1)) ^ 2) / C2
2070 P2 =  - (A1 * ( SIN (C1)) ^ 2 + B1 * ( COS (C1)) ^ 2) / C2
2080 P3 = (2 *  COS (C1) *  SIN (C1) * (B1 - A1)) / C2
2090  IF H(H2,7) > H(H2,8) THEN CR =  LOG ((H(H2,7) - H(H2,8)) / H(H2,7))
2100 QI = P1 * XS ^ 2 + P2 * YS ^ 2 + P3 * XS * YS
2110 H(H2,0) = 1
2120  IF QI < CR THEN 2150
2130 FI = H(H2,3) + H(H2,7) * ( EXP (QI) - 1)
2140  IF FI > Z THEN Z = FI
2150  NEXT
2160  RETURN
```

```
//PRO9   JOB (3102,1087),'HOCK SMC1013'
// EXEC SIM25CLG
//SIM.SYSIN DD *
PREAMBLE
     THE SYSTEM OWNS A SHOTLIST
     THE SYSTEM OWNS A CASLIST
PERMANENT ENTITIES
     DEFINE VEHCOUNT, MAXBLUE, MAXRED AS INTEGER VARIABLES
     DEFINE BLUE.TIME, RED.TIME, TOTAL.TIME AS REAL VARIABLES
     DEFINE TOTAL.CASTIME AS A REAL VARIABLE
TEMPORARY ENTITIES
     GENERATE LIST ROUTINES
     EVERY SHOT HAS A TTT, A TIME, A RANGE, A FNAME, A FSWTYPE,
                    A TNAME, A TSWTYPE, A STATUS,
                    AND BELONGS TO A SHOTLIST
     EVERY CAS HAS A NUMCAS, A TIMECAS, A TGT.CAS,
                    A TYPE.CAS, A STATUS.CAS, A T.BTWN.CAS,
                    AND BELONGS TO A CASLIST
DEFINE SHOTLIST AS A SET RANKED BY LOW TIME
     DEFINE TTT, RANGE, TNAME, FNAME AS INTEGER VARIABLES
     DEFINE FSWTYPE, TSWTYPE, STATUS AS ALPHA VARIABLES
     DEFINE TIME, T.BTWN.CAS AS INTEGER VARIABLES
     DEFINE NSHOT AS AN INTEGER VARIABLE
DEFINE CASLIST AS A SET RANKED BY LOW TIMECAS
     DEFINE T.CAS, NUMBL.CAS, NUMRD.CAS AS INTEGER VARIABLES
     DEFINE NUMCAS, TIMECAS, TGT.CAS AS INTEGER VARIABLES
     DEFINE TYPE.CAS, STATUS.CAS AS ALPHA VARIABLES
DEFINE VAR1, VAR2, VAR3, VAR4, VAR5, VAR6 AS REAL VARIABLES
     DEFINE MINP, MINC, PHAT, QHAT, AHAT, BHAT, P AS REAL VARIABLES
     DEFINE LP,LQ AS REAL VARIABLES
     DEFINE CK, K, CR, CB AS INTEGER VARIABLES
     DEFINE MK1, NK1 AS INTEGER VARIABLES
     DEFINE RUNNO, TMCAS, CURTIME AS INTEGER VARIABLES
     DEFINE TBLUE, TRED AS INTEGER VARIABLES
     DEFINE TMBLCAS, TMRDCAS AS INTEGER VARIABLES
ACCUMULATE NSHOT AS THE NUMBER OF TNAME
TALLY J.BL AS THE NUMBER, BLMU AS THE MEAN,BLSIG AS THE STD.DEV,
AND BLHST(0 TO 200 BY 10) AS THE HISTOGRAM OF BLUE.TIME
TALLY J.RD AS THE NUMBER, RDMU AS THE MEAN, RDSIG AS THE STD.DEV,
AND RDHST(0 TO 200 BY 10) AS THE HISTOGRAM OF RED.TIME
TALLY JTOT AS THE NUMBER,TOTMU AS THE MEAN,TOTSIG AS THE STD.DEV,
AND TOTHST(0 TO 200 BY 10) AS THE HISTOGRAM OF TOTAL.TIME
TALLY JCAS AS THE NUMBER, CASMU AS THE MEAN, CASIG AS THE STD.DEV,
AND CASHST (0 TO 1600 BY 100) AS THE HISTOGRAM OF TOTAL.CASTIME
END
MAIN
DEFINE I AS AN INTEGER VARIABLE
DEFINE II AS AN INTEGER VARIABLE
```

```
READ RUNNO,VEHCOUNT,MAXBLUE,MAXRED
   PRINT 1 LINE WITH RUNNO THUS
    ++++++++++   RUN NUMBER ****** ++++++++++
   SKIP TWO LINES
PRINT 1 LINE WITH MAXBLUE,MAXRED THUS
       INITIAL BLUE STRENGTH =  ****  INITIAL RED STRENGTH = ****
   SKIP TWO LINES
     USE UNIT 12 FOR INPUT
     LET I = 0
'NEW'
     CREATE A SHOT
'READ'
     IF DATA IS ENDED, GO TO 'PROCESS' ELSE START NEW RECORD
     IF MODE IS NOT INTEGER SKIP 1 RECORD GO TO 'READ' ELSE
       SKIP 4 FIELDS READ FSWTYPE(SHOT)
       IF FSWTYPE(SHOT) EQUALS "EXPD" GO TO 'READ2' ELSE
       LET RCOLUMN.V=1
     READ TTT(SHOT), TIME(SHOT), RANGE(SHOT), FNAME(SHOT),
          FSWTYPE(SHOT)
     SKIP 1 FIELD READ TNAME(SHOT),TSWTYPE(SHOT)
     SKIP 12 FIELDS READ STATUS(SHOT)
       IF FSWTYPE(SHOT) IS EQUAL TO "2-4" OR FSWTYPE(SHOT) = "2-8"
       DESTROY THIS SHOT LET RCOLUMN.V=132 GO TO 'NEW' ALWAYS
       IF FSWTYPE(SHOT) IS EQUAL TO "3-6" OR FSWTYPE(SHOT) = "4-1"
       DESTROY THIS SHOT LET RCOLUMN.V=132 GO TO 'NEW' ALWAYS
  FILE SHOT IN SHOTLIST
    LET RCOLUMN.V=132
   GO TO 'NEW'
'READ2'
     LET RCOLUMN.V=1
     SKIP 2 FIELDS READ TIME(SHOT) SKIP 1 FIELD READ TNAME(SHOT),
            TSWTYPE(SHOT)
     SKIP 5 FIELDS READ STATUS(SHOT)
      FILE SHOT IN SHOTLIST
     LET RCOLUMN.V=132
       GO TO 'NEW'
'PROCESS'
''   LIST ATTRIBUTES OF EACH SHOT IN SHOTLIST
''   PRINT 1 LINE WITH NSHOT THUS
''   THE NUMBER OF SHOTS IN THE LIST = ****
''    THIS PORTION OF PROGRAM FINDS THE CASUALTIES
'NEWCAS'
      FOR EACH SHOT IN SHOTLIST, DO
          IF STATUS(SHOT) IS NOT EQUAL TO "DEAD" GO TO 'LOOP' ALWAYS
              ADD 1 TO T.CAS
        IF TNAME(SHOT) IS GREATER THAN 41 ADD 1 TO NUMRD.CAS ELSE
                                          ADD 1 TO NUMBL.CAS ALWAYS
      CREATE A CAS
```

```
          LET NUMCAS(CAS) = T.CAS
          LET TIMECAS(CAS) = TIME(SHOT)
              LET TOTAL.CASTIME = TIMECAS(CAS)
;;            PRINT 1 LINE WITH TOTAL.CASTIME THUS
;;              TOTAL.CASTIME = ****.****
          IF T.CAS = 1 LET T.BTWN.CAS(CAS) = 0       ELSE
          LET T.BTWN.CAS(CAS) = TIMECAS(CAS)-TMCAS
          LET TOTAL.TIME = T.BTWN.CAS(CAS)           ALWAYS
              IF T.CAS = 1 RESET THE TOTALS OF TOTAL.TIME   ALWAYS
;;     PRINT 1 LINE WITH TOTAL.TIME THUS
;;       TOTAL.TIME = *****.******
          LET TMCAS = TIMECAS(CAS)
          LET TGT.CAS(CAS) = TNAME(SHOT)
          LET TYPE.CAS(CAS) = TSWTYPE(SHOT)
             IF TNAME(SHOT) IS GREATER THAN 41 GO TO 'RED'   ELSE
               LET STATUS.CAS(CAS) = "BLUE"
               LET BLUE.TIME = TIMECAS(CAS)-TMBLCAS
               LET TMBLCAS = TIMECAS(CAS)
;;             PRINT 1 LINE WITH BLUE.TIME THUS
;;              BLUE.TIME = ******.*****
               IF NUMBL.CAS = 1 RESET THE TOTALS OF BLUE.TIME  ALWAYS
          FILE CAS IN CASLIST
          GO TO 'LOOP'
'RED'     LET STATUS.CAS(CAS) = "RED"
          LET RED.TIME = TIMECAS(CAS)-TMRDCAS
          LET TMRDCAS = TIMECAS(CAS)
;;        PRINT 1 LINE WITH RED.TIME THUS
;;        RED.TIME = ******.******
          IF NUMRD.CAS = 1 RESET THE TOTALS OF RED.TIME ALWAYS
          FILE CAS IN CASLIST
'LOOP'    LOOP
     LET NUMBL.CAS = T.CAS - NUMRD.CAS
;;   PRINT 1 LINE WITH T.CAS, NUMBL.CAS, NUMRD.CAS THUS
;; T.CAS= *****  NUM BLUE CAS = *****  NUM RED CAS = *****
;;   LIST ATTRIBUTES OF EACH CAS IN CASLIST
     PRINT 4 LINES THUS

    SHOT LIST DATA FOR CASUALTIES FOR THIS STAR SIMULATION

     NUM  TIME   TARGET  SYS-WPN TYPE  TIME BETWEEN CASUALTIES
FOR EACH CAS IN CASLIST, PRINT 1 LINE WITH NUMCAS(CAS),TIMECAS(CAS),
     TGT.CAS(CAS), TYPE.CAS(CAS), T.BTWN.CAS(CAS) THUS
     ***  ****    ***       ***                 *********
     SKIP 2 LINES
     PRINT 2 LINES WITH NUMBL.CAS, BLMU, BLSIG,
                        NUMRD.CAS, RDMU, RDSIG THUS
  NUMBER OF BLUE KILLED = *****  BLMU =  ***.***** BLSIG = ***.***
  NUMBER OF RED KILLED =  *****  RDMU =  ***.***** RDSIG = ***.***
```

```
      PRINT 4 LINES THUS

     HISTOGRAM OF BLUE AND RED CASULTIES
       (DATA BASED CN TIME BETWEEN BLUE CASULTIES
        AND TIME BETWEEN RED CASULTIES)
      PRINT 2 LINES THUS

     INTERVAL=10 UNITS    #BLUE CAS  #RED CAS
     FOR II = 1 TO 21, PRINT 1 LINE WITH II,BLHST(II),RDHST(II) THUS
            **                  ****       ****

SKIP 2 LINES
     PRINT 1 LINE WITH T.CAS,TOTMU,TOTSIG THUS
    TOTAL CASUALTIES = *****        TOTMU = ****.****  TOTSIG = ***.***
     SKIP 2 LINES
     PRINT 3 LINES THUS
     HISTOGRAM OF TIME BETWEEN CASUALTIES(TOTAL)

     INTERVAL=10 UNITS     #CASULTIES
     FOR I = 1 TO 21, PRINT 1 LINE WITH I,TOTHST(I) THUS
           **                  ****
SKIP 2 LINES
     PRINT 3 LINE THUS
      THE FOLLOWING DATA IS BASED ON CASUALTIES AS
      THEY OCCUR ON THE TIME LINE (I.E. THE NUMBER
      OF CASUALTIES THAT OCCUR IN THE FIRST 100 UNITS)
SKIP 2 LINES
      PRINT 1 LINE WITH T.CAS, CASMU, CASIG THUS
      TOTAL CASUALTIES =  ****   CASMU = ****.****   CASIG = ****.****
      PRINT 2 LINES THUS

      HISTOGRAM OF TOTAL CASUALTIES AS A FUNCTION OF TIME PERIOD
      PRINT 2 LINES THUS

      INTERVAL=100 UNITS  #CASUALTIES
     FOR I = 1 TO 17, PRINT 1 LINE WITH I,CASHST(I) THUS
           ****            ********
       LET MINP=1.0  LET MINQ=1.0  LET PHAT=1.0  LET QHAT=1.0
      FOR P = .010 TO .99 BY .01, DO
          LET NK1 = MAXRED LET MK1 = MAXBLUE
          LET VAR1 = 0  LET VAR2 = 0  LET VAR3 = 0
          LET VAR4 = 0  LET VAR5 = 0  LET VAR6 = 0
          FOR EACH CAS IN CASLIST, DO
!!       PRINT 1 LINE WITH STATUS.CAS(CAS), T.BTWN.CAS(CAS) THUS
!!             STATUS.CAS(CAS) = *****    TIME = ********
          IF NUMCAS(CAS) = 1 CYCLE ALWAYS
             IF STATUS.CAS(CAS) = "RED"  LET CK=0
                                LET NK1=NK1-1  ALWAYS
```

```
            IF STATUS.CAS(CAS) = "BLUE" LET CK=1
                                  LET MK1=MK1-1  ALWAYS
!!        PRINT 1 LINE WITH NK1,MK1,CK,P THUS
!!          NK1 = ******     MK1 = *****    CK = ** P = *.****
    LET VAR1=VAR1+((1-P**MK1)*NK1)*(T.BTWN.CAS(CAS))
    LET VAR2=VAR2+(MK1*P**(MK1-1))*NK1*(T.BTWN.CAS(CAS))
    LET VAR3=VAR3+(CK*MK1*P**(MK1-1))/(1-P**MK1)
    LET VAR4=VAR4+(1-P**NK1)*MK1*(T.BTWN.CAS(CAS))
    LET VAR5=VAR5+(NK1*P**(NK1-1)*MK1*T.BTWN.CAS(CAS))
    LET VAR6=VAR6+((1-CK)*NK1*P**(NK1-1))/(1-P**NK1)
    LOOP
!! PRINT 2 LINES WITH VAR1,VAR2,VAR3,VAR4,VAR5,VAR6 THUS
!! VAR1=***.***  VAR2=***.*** VAR3=***.***
!! VAR4=***.***  VAR5=***.*** VAR6=***.***
!!    LET LP=TBLUE/VAR1-VAR3/VAR2
      LET LP=NUMBL.CAS/VAR1-VAR3/VAR2
            IF ABS.F(LP) < MINP  LET MINP = ABS.F(LP)
            LET PHAT = P  LET AHAT = NUMBL.CAS/VAR1     ALWAYS
      LET LQ=NUMRD.CAS/VAR4-VAR6/VAR5
            IF ABS.F(LQ) < MINQ  LET MINQ = ABS.F(LQ)
            LET QHAT = P LET BHAT=NUMRD.CAS/VAR4        ALWAYS
!! PRINT 1 LINE WITH NK1, MK1 THUS
!!   NK1=*****   MK1=*****
    PRINT 1 LINE WITH PHAT,QHAT,AHAT,BHAT THUS
       PHAT= ****.**** QHAT= ****.**** AHAT= ****.**** BHAT= ****.***
    PRINT 1 LINE WITH LP,LQ THUS
      LP = *****.*********    LQ = *****.*********
    LOOP
STOP END
//GO.SIMU06 DD DISP=SHR,VOL=SER=MVS004,UNIT=3350,DSN=S3102.TRANFILE
//GO.SIMU12 DD DISP=SHR,DSN=MSS.S3102.THESIS(TABLE6)
//GO.SYSIN DD *
    59 186 12 30
```

LIST OF REFERENCES

1. Hardison, D. C., and others, Special Study Group, Department of the Army, Review of Army Analysis, Volume I - Main Report, section 3-4, April, 1979.

2. Lucas, G., "Generalized Differential Equations of Combat", A Syllabus of Models for Economic, Personnel and Force Effectiveness Studies, p.262, 1 December 1971.

3. Ibid., p. 263.

4. Clark, G. M., The Combat Analysis Model, Ph.D. Thesis, p. 14, The Ohio State University, Columbus, Ohio, 1969.

5. Command and Control Technical Center, VECTOR-2 System for Simulation of Theater-level Combat, TM 201-79, pp. 51-55 and pp. 68-71, Washington D.C., 16 January 1979.

6. Clark, G. M., The Combat Analysis Model, Ph.D. Thesis, p. 149, The Ohio State University, Columbus, Ohio, 1969.

7. Taylor, J. G., Force-on Force Attrition Modelling, Military Applications Section of the Operations Research Society of America, p. 29, Arlington, Virginia, 1980.

8. Ibid., p. 33.

9. Ibid., p. 77.

10. Andreghetti, J., A Model for the Statistical Analysis of Land Combat Simulation and Field Experimentation Data, M.S. Thesis, pp. 66-103, Naval Postgraduate School, Monterey, CA, September, 1973.

11. Clark, G. M., The Combat Analysis Model, p. 179, Reprinted from Proceedings 24th Military Operations Research Symposium, November, 1969.

12. Clark, G. M., The Combat Analysis Model, Chapter 11 in "The Tank Weapon System", A.B. Bishop and G.M. Clark (Editors), Report No. AR 69-2B, p. 11-54, System Research Group, The Ohio State University, Columbus, Ohio, 1969.

13. Apple Computer Inc., The DOS Manual-Disk Operating System, 1980.

14. Hartman, J. K., Parametric Terrain and Line of Sight Modelling in the STAR Combat Model, Naval Postgraduate School, Monterey, CA, unpublished paper.

BIBLIOGRAPHY

Apple Computer Inc., The Applesoft Language Manual, 1978.

Kiviat, P. J., Villanueva, R. and Markowitz, H.M., SIMSCRIPT II.5 Programming Language, 2nd Edition, Consolidated Analysis Centers, Inc., 1973.

Needels, C. J., Parameterization of Terrain in Army Combat Models, M.S. Thesis, Naval Postgraduate School, Monterey, March 1976.

Taylor, J. G., Lanchester-type Models of Warfare, draft edition, Volume 2, Naval Postgraduate School, Monterey, CA, 1980.

Taylor, J. G., Preliminary Investigation of Improved Attrition Methodolgy for Aggregated-Force Ground-combat Models, Naval Postgraduate School, Monterey, CA, 30 June 1981.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Chief TRADOC Research Element Monterey Naval Postgraduate School Monterey, California 93940	1
5.	Associate Professor James K. Hartman Code 55Hh Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	Associate Professor S. H. Parry Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	5
7.	Headquarters US Army Training and Doctrine Command ATTN: Director, Studies and Analysis Directorate, Mr. S. Goldberg Fort Monroe, Virginia 23651	1
8.	Associate Professor A. L. Schoenstadt Code 53Zh Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
9.	Professor James G. Taylor Code 55Tw Department of Operations Research Naval Postgraduate School Monterey, California 93940	10

10. Office of the Commanding General 1
US Army Training and Doctrine Command
ATTN: General Glenn Otis
Fort Monroe, Virginia 23651

11. Headquarters 1
US Army Training and Doctrine Command
ATTN: ATCG-T (BG Morelli)
Fort Monroe, Virginia 23651

12. Office of the Commanding General 1
US Readiness Command
ATTN: General Donn A Starry
MacDill AFB, Florida 33621

13. Mr. Walter Hollis 1
Deputy Under Secretary of the Army
(Operations Research)
Department of the Army, The Pentagon
Washington, D. C. 20310

14. LTG Howard Stone 1
Commanding General
US Army Combined Arms Center
Fort Leavenworth, Kansas 66027

15. Director 1
Combined Arms Combat Development Activity
ATTN: ATZL-CAC-A (Mr. Lee Pleger)
Fort Leavenworth, Kansas 66027

16. Director 1
Combat Analysis Office
ATTN: Mr. Kent Pickett
US Army Combined Arms Center
Fort Leavenworth, Kansas 66027

17. Command and General Staff College 1
ATTN: Education Advisor
Room 123, Bell Hall
Fort Leavenworth, Kansas 66027

18. Dr. Wilbur Payne, Director 1
US Army TRADOC Systems Analysis Activity
White Sands Missile Range, New Mexico 88002

19. Headquarters, Department of the Army 1
Office of the Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-2D
Washington, D.C. 20310

20. Commander 1
US Army Concepts and Analysis Agency
ATTN: MOCA-WG
8120 Woodmont Avenue
Bethesda, Maryland 20014

21. Director 1
US Army Material Systems Analysis Activity
ATTN: DRXSY-CM (Mr. Bill Niemeyer)
Aberdeen Proving Grounds, Maryland 21005

22. Director 1
US Army Night Vision and Electro-Optical Lab
ATTN: DEL-NV-VI (Mr. Frank Shields)
Fort Belvoir, Virginia 22060

23. Director 1
USATRASANA
ATTN: Mr. Ray Heath
White Sands Missile Range, New Mexico 88002

24. Director 1
Combat Developments, Studies Division
ATTN: MAJ W. Scott Wallace
US Army Armor Agency
Fort Knox, Kentucky 40121

25. Commandant 1
US Army Field Artillery School
ATTN: ATSA-CD-DSWS
Fort Sill, Oklahoma 73503

26. Director 1
Combat Developments
US Army Aviation Agency
Fort Rucker, Alabama 36362

27. Director 1
Combat Developments
US Army Infantry School
Fort Benning, Georgia 31905

28. Director 1
Combat Developments
ATTN: ATZA-CDE (CPT James Mudd)
US Army Engineer School
Fort Belvoir, Virginia 22060

29. Director 1
Combat Developments
ATTN: ATSA-CDF-S
US Army Air Defense Agency
Fort Bliss, Texas 79905

30. Commander 1
US Army Logistics Center
ATTN: ATCL-OS (Mr. Cammeron/CPT McGrann)
Fort Lee, Virginia 23801

31. Director 1
Combat Developments
ATTN: LTC Harnagel
US Army Signal School
Fort Gordon, Georgia 30905

32. Deputy Under Secretary of the Army 1
for Operations Research
Room 2E261, Pentagon
Washington, DC 20310

33. MAJ Jeffrey L. Ellis, USA 1
Department of Operations Research, Code 55EI
Naval Postgraduate School
Monterey, California 93940

34. CPT Charles J. Mckenzie III 1
US Readiness Command
MacDill AF Base, Florida 33621

35. CPT Thomas J. Pawlowski III 1
Department of Mathematical Sciences (DFMS)
US Air Force Academy, Colorado 80840

36. Mr. Ernest Boehner 1
CACDA/CASAA
Ft. Leavenworth, Kansas 66027

37. Mr. David Goodwin 1
1301 Stillhouse Creek Rd
Chesterfield, Missouri 63017

38. CPT James V. Mudd 1
35 Heussy Ave.
Buffalo, New York 14220

39. CPT Bruce Hock 1
288 Alexander Drive
Manchester, New Hampshire 03103

40. CPT William T. Farmer 1
TRADOC Research Element Monterey
Naval Postgraduate School
Monterey, California 93940

198092
Thesis
M2234 McKenzie
c.1 Deterministic
aggregated model
of STAR on the apple
computer (DAMSTAC).